DIFFERENCES BETWEEN CONTINUOUS AND DISCRETE FORMULATIONS OF DELAY PROCESSES

A THESIS

Presented to

The Faculty of the Division of Graduate Studies

By

David Harvey Vatz

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in the School of

Industrial and Systems Engineering

Georgia Institute of Technology

June, 1976

DIFFERENCES BETWEEN CONTINUOUS AND DISCRETE

FORMULATIONS OF DELAY PROCESSES

Approved:

Richard Wright, Chairman

Stuart Deutsch

I. Gayden Thompson

Date approved by Chairman: May 27, 1976

ACKNOWLEDGMENTS

I wish to express my deepest appreciation to my advisor, Dick Wright, for his many hours of advice and assistance without which this thesis would not have been completed.

I am also grateful to the other members of my reading committee, Drs. Stuart Deutsch and Gayden Thompson, for their constructive criticism which helped direct the final form of the manuscript.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

SUMMARY

A delay process is a substructure that accepts elements of an inflow and holds them for some time lag. Such a process includes both an outflow and an internal accumulation. Delay processes appear in all real world dynamic systems. Continuous simulation models exploit simple state-space structures to approximate real world delay processes. This thesis investigates how closely the output characteristics of state space delay approximations match output from their real world referents.

Output characteristics of state space delays are analytically tractable and easily summarized. The behavior of real world delays is not analytic; the thesis derives descriptors of the output -- mean, variance, autocorrelation, attenuation and recognition. Results are offered for delay processes with inputs that are constant, sinusoid and white noise, and for lag time distributions that are uniform, exponential, Erlang3 or normal.

The absolute differences between discrete and continuous processes are shown to depend on the fundamental time unit of the system, the magnitude of the input flow, and the lag time distribution. The relative adequacy of a continuous surrogate depends on the nature of the system in which it is embedded. Schemes for improving the output characteristics of continuous delays are developed.

CHAPTER I

INTRODUCTION

This chapter defines the objective of the thesis, describes the scope and significance of the problem addressed, presents the methodology used, and outlines the presentation of results.

1.1 Objective and Scope

The objective of this thesis is to give a quantitative comparison of the performance of discrete delay processes found in real systems with the performance of surrogate delay components used in continuous simulation models.

The scope of this thesis is limited to a particular category of continuous surrogate delays and also to a particular class of discrete delay processes. Only linear state space surrogates, the so called "nth order exponential delays" are considered; but these structures are used in the majority of continuous simulation models. Only the class of discrete real world delays where entity entry and exit occur at equally spaced time intervals is investigated. This restricted class of discrete delays allows tractable mathematical derivations but is general enough to include those processes which a state space modeler would seek to approximate with a continuous surrogate.

1.2 The Problem

A real world delay process is a substructure that accepts elements of an inflow and holds them for some probabilistic time lag. State space surrogates use state variables and differential equations to approximate the behavior of discrete delays. State space delays do not perfectly mime real world delay processes. The major source of error is the failure of state space surrogates to treat flows as streams of discrete entities. That is, the state space approach assumes the input and output flows to be composed of infinitesimal units, such as in the flow of electricity.

Two further problems need to be noted. State space delays behave deterministicly; they break each input pulse into a train of output pulses according to an impulse response function. This is in contrast with the random selection of lag times in discrete delays. Finally, the expected behavior of state space surrogates cannot always be equated to the expected behavior of the referent real world delay. State space delays are restricted to Erlang-shaped impulse response functions, whereas real world delay's lag time distributions can be of any form.

1.3 Significance of the Problem

State space delay processes are used as components of continuous models. Any inaccuracy in the design of the delay surrogate impacts the accuracy of the total model. The adequacy of these models depends in part on the degree of inaccuracy in the delay process' formulation and also on the sensitivity of the particular model to that inaccuracy. There has been very little mention of this potential model inaccuracy from state space delay surrogates. Further, continuous simulation techniques as embodied in DYNAMO and GASP IV are being used for an expanding variety of discrete real world problems. Hence it is important to provide quantitative analysis of the discrepancy between the performance of state space surrogates and the performance of the real world discrete delays.

1.4 Methodology

The thesis compares discrete delay performance with the performance of the continuous delay surrogate for several forms of input; constant, sinusoidal and white noise. The methodology adopted here comprises four stages. The first stage defines the behavior of the continuous state space delay. State space behavior is analytic and is summarized from the continuous simulation literature. Second, the thesis derives equations describing the performance of discrete real world delays. This derivation is necessary since similar equations cannot be found in the literature. The third stage identifies delay descriptors, such as delay lag distribution, which importantly affect the behavior of delay process performance. And finally, the sensitivity of delay performance to changes in values of delay descriptors is analyzed.

1.5 Presentation of Results

The remainder of the thesis includes five chapters. Chapter Two provides background information essential to the thesis development. The major sections define "delay process", describe the continuous and discrete modeling formulations of delay processes, survey available literature and list the generic comparisons which the thesis addresses.

Chapters Three and Four provide tools for quantitative analysis of performance differences between modeling formulations. Chapter Three develops the analysis of discrete versus continuous formulations under the condition of constant input. That chapter contrasts formulation behavior using output mean, variance and autocorrelation as measures of comparison. Several important delay descriptors are proposed and their impact on performance derived. These descriptors are input magnitude, delay distribution shape, delay distribution width, and the system's fundamental sampling time.

Chapter Four considers performance differences for sinusoidal input and for white noise input. The performance measure for sinusoidal input is attenuation. The impact of carrier magnitude, sinusoid frequency, delay shape and delay width on attenuation are described. The performance measure for white noise is the variance gain, the ratio of output variance to input variance. The impacts of input magnitude, input variance, delay shape, and delay width on this measure are derived.

Chapter Five offers an example of the impact of delay component error on overall model adequacy. A specific inventory-production system is used. That system which includes a single first order exponential delay is simulated using both discrete and continuous formulations. The performance of the two models is compared yielding a quantitative measure of the impact of delay process formulation on model performance. The chapter then illustrates a series of fixes which can be incorporated into a naive continuous model to improve its performance. The fixes add white noise element to the model; white noise added to the outflow of a state space delay process is seen to be an effective improvement.

CHAPTER II

BACKGROUND INFORMATION

Chapter Two provides background information to support the analysis presented in future chapters. The chapter begins by defining "delay process" in section 2.1. Alternative simulation formulations are discussed in section 2.2, describing how discrete and continuous simulation approaches incorporate delay processes into models. A survey of available literature is provided in section 2.3. Section 2.4 defines important variables and delineates the major issues which the thesis addresses.

2.1 Delay Processes Defined

Before any comparison of discrete versus continuous formulations of delay processes can be performed, it is essential to understand just what a delay process is. This section defines the structure and output of delay processes and provides examples of delay processes in everyday life.

2.1.1 The Structure of a Delay Process

A delay process is a component which accepts elements of an inflow and holds them for some time lag. Such a process includes both an outflow and an internal accumulation. The delay process is discrete, probabilistic and conservative. Each discrete entity in the input stream individually draws an "in-process" duration from some distribution of lag times. The process is conservative: all entering entities are eventually emitted after residing in the accumulation for their respective lag durations.

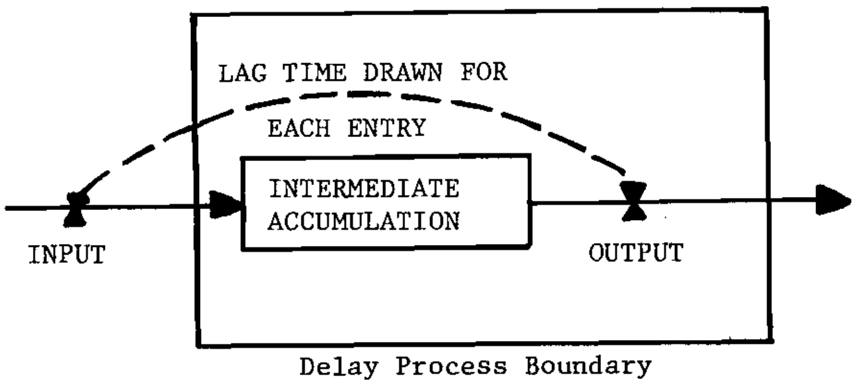


Figure 1. Delay Process

Figure 1 illustrates that the delay mechanism is independent of the input stream. The delay mechanism is totally defined by the probability distribution (also called "delay distribution") from which the "in-process" lags are drawn. Two important descriptors of the delay distribution are 1) mean delay time and 2) distribution shape.

2.1.2 Output from a Delay Process

The output from a delay process depends on both the nature of the input series and the structure of the delay distribution. Consider a deterministic delay process where each entity resides in the intermediate accumulation for precisely M time units, then exits. Entities which enter the accumulation as a group, leave the accumulation as a group. Figure 2 illustrates the output of this deterministic delay process with input pulses of 1, 4 and 2 units.

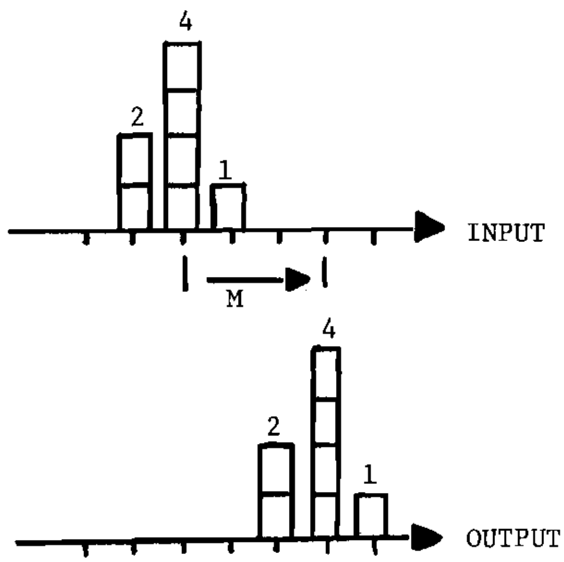


Figure 2. Output of Deterministic Delay Process

Now consider a delay process categorized by a delay distribution of exponential shape and mean M. Each arriving entity draws a sample from the delay distribution, and resides in the accumulation for the indicated duration. The dispersion of draws from the delay distribution distorts any pattern of the input stream. For example, fifteen entities entering the delay at a single time period may leave in a fashion shown in Figure 3.

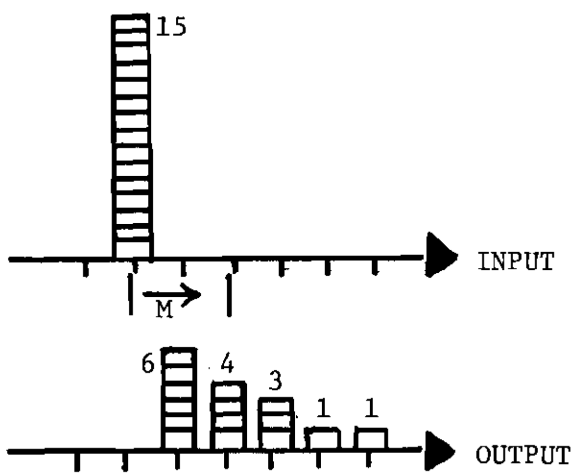


Figure 3. Output of Delay Process with Dispersion

2.1.3. Occurrence of Delay Processes

We have considered a delay process (here termed a "delay") as a distinct element. In fact these structures are usually a component of larger systems. In such a case, the delay controls entity movement across an input-output boundary. In the figure below, the delay controls the movement of entities from point "a" to point "b".

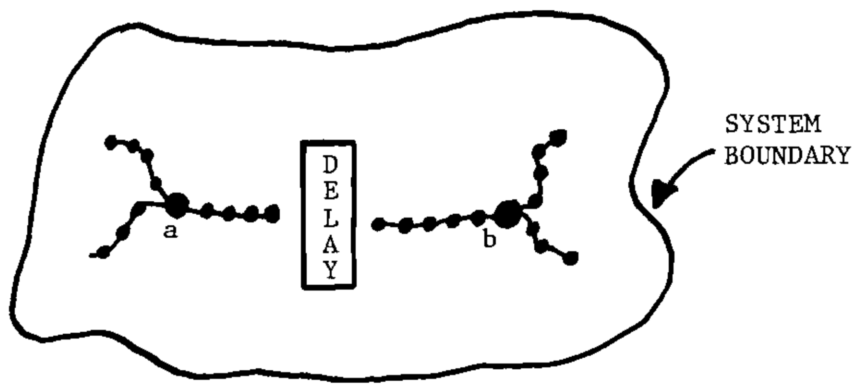


Figure 4. Delay as a Component of a System

Were the delay in Figure 4 removed, entity movement from point "a" to point "b" would occur instantaneously.

Two examples may be helpful. Walking to lunch can be viewed as a delay process with people in transit as the intermediate accumulation and walking time as the "in-process" lag. One enters the delay by leaving the office building; one exits by arriving at the lunch cafeteria. As each person walks at his own rate and uses a particular route to the cafeteria, there is a distribution of walking times.

A second example includes a music listening room. Suppose that all music played in the room is preselected by the management. People who enter the room (since they have no choice) do not queue or wait to hear a favorite song. They merely remain for as long as they desire. The intermediate accumulation in this example is the people in the room. The lag distribution is the distribution of time that browsers spend in the room. In both of these examples, the delay process acts upon entities in an individualistic and probabilistic manner.

2.2 Delays in Simulation Models

Two different conceptual approaches are used in simulation to model delays. The distinction is in extent of aggregation of flows. Discrete, or better named item-by-item simulation considers individual entities (e.g., job number 136 arriving; person John Doe migrating). Discrete simulation models maintain detailed file structures noting the status of individual entities. Continuous simulation models aggregate entities into continuous flows (e.g., the rate of jobs per week; or migrants per year). Continuous models use differential or difference equations to describe system status. This section illustrates how delay processes are incorporated into each simulation approach.

2.2.1 Delays in Discrete Simulation

In discrete simulation, representing delays is straightforward. The formulation mimics directly the referent system. Each entity flowing into a delay is assigned a random draw determining lag duration. The entity then joins an accumulation file. The output from the delay at time T is equal to the number of entities marked to exit the accumulation file at that time.

Straightforward programming techniques incorporate discrete delays in simulation models[1]. Filing systems to hold and access items of in-process accumulations are available as are "process generators" to yield draws from lag distributions of analytic or approximation form[2].

To exemplify the operation of a discrete delay process we return to the music browsing room. Assume that at clock time 420 three people enter the room. The simulation program assigns each person an identification number (say: 101, 102 and 103), and determines an exit time for each person by making a draw from an appropriate pseudorandom process-generator. If the in-process lag times are uniform on (10,30), we may see a simulation status dump of:

Time: 420

Person Number	99	100	101	102	103
Location	Room	Room	Room	Room	Room
Time of Next Transition	443	421	438	431	439

Figure 5. Status Dump After Arrival

The status dump shows that the three entities join a file of entities already in the accumulation (room). The new entrants selected draws of 18, 11, and 19 time units and will exit at time 438, 431, and 439 respectively. To determine which entity is the next to leave the delay (room), the simulation logic scans over all entries in the file, and locates the minimum time of transition, the next event. In the case above, the next event occurs at time 421 and happens to person 100 leaving

---

1 For an example of discrete simulation with careful exposition of file structures and discrete delays, see Thompson [1974].

2 See Fishman [1973].

the room.

To execute the next event, the clock moves to 421 and person 100 exits. This is done by filling the storage locations formerly allocated to person 100 with nonsense values such as 99999 noting that the locations are not in use. A status dump of the list after the departure of person 100 would show:

Person Number	99	99999	101	102	103
Location	Room	Gone	Room	Room	Room
Time of Next Transition	443	99999	438	431	439

Figure 6. Status Dump After Departure

In a discrete model, as long as an entity is "in the system" it is recorded in a file structure such as the above. This treatment of real world discrete delays is accurate, but can be very expensive in terms of model execution time. And as the number of entities moving through the delay grows, so does model execution time.

In continuous simulations the opposite tradeoff is made. Execution time of a continuous delay is independent of the number of entities within the delay, but entity processing is approximate. Thus, for systems with many moving entities, an analyst may choose to model a discrete process with a continuous delay.

2.2.2 Delays in Continuous Simulation

In continuous flow simulation delays are represented with state space mechanisms. A continuous delay is a set of differential equations, one equation needed for each intermediate accumulation. As we will see, often phantom (non-casual) intermediate accumulations are included in the continuous modeling artifact.

2.2.2.1 First Order Delay. The simplest continuous delay[1] uses a single accumulation, and is termed a "1st order exponential delay" (see Figure 7).

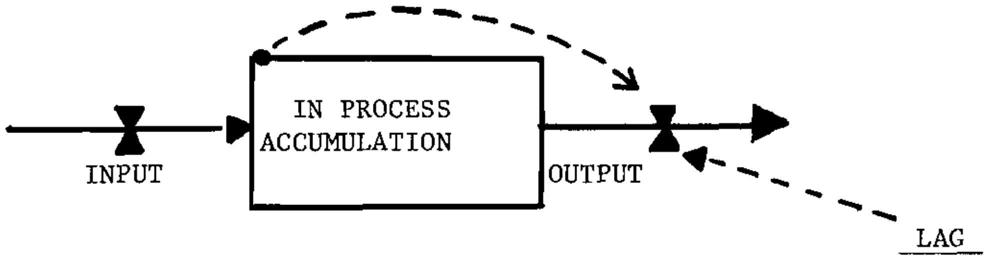


Figure 7. First Order Exponential Delay

The above structure responds to a pulse input as shown in Figure 8.

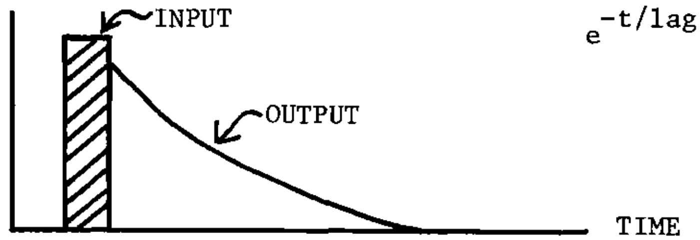


Figure 8. Impulse Response: First Order Exponential Delay

It is easy to show that the average transit time for an entering pulse is equal to "lag" and further, the outflow follows an exponential law:

$$\text{Outflow } (t) = \frac{e^{(-1/lag)t}}{lag} \tag{2-1}$$

[1] "State space delay" and "exponential delay" will be used synonymously with "continuous delay."

where t is the time after pulse entrance.

2.2.2.2 Higher Order Delays. To create different impulse response functions, phantom accumulations are inserted in the delay surrogate. The "order" of an exponential delay is the number of accumulations (hence, number of differential equations) in the structure. The extension of the structure of Figure 7 yields the general linear exponential delay. Linear exponential delays are described by a set of linear differential equations of a particular format:

$$\dot{X} = AX + Bu \qquad (2\text{-}2)$$
$$y = CX$$

Here X is a vector of the contents of the accumulations of the delay, A is an n by n matrix, B and C are vectors, and y and u are the scalar output and input respectively. More particularly for the nth order exponential delay, A, B, and C, must follow the pattern below:

$$A = \begin{bmatrix} \frac{n}{D} & 0 & 0 & \cdots & 0 \\ \frac{n}{D} & -\frac{n}{D} & 0 & \cdots & 0 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & 0 & 0 & \cdots & \frac{n}{D} \; -\frac{n}{D} \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \vdots \\ \cdot \end{bmatrix} \quad C = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \frac{n}{D} \end{bmatrix} \qquad (2\text{-}3)$$

In equation (2-3) n is the order of the delay, and D is the average lag time for the entire process.

The third order exponential delay, which is easily realized (using three differential equations) has an Erlang3 shaped impulse response. It is formed from three serial first order exponential delays. To produce total average lag of D, each first order delay has average lag duration of D/3 (see Figure 9). It is often used by continuous simulators to approximate Gaussian dispersions and the no dispersion pipeline delay.

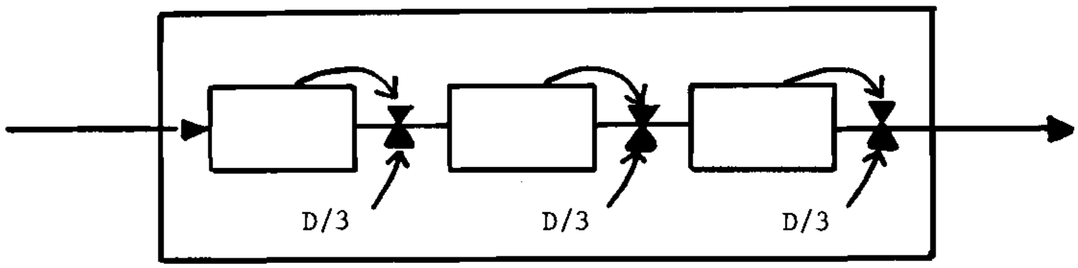


Figure 9. Third Order Delay

The impulse response function for various exponential delays appear in Figure 10. As the order of the delay grows, the impulse response function tends to be more symmetric, and the mode of the function grows closer to the mean. The infinite order delay has an impulse response function equal to that of a determinsitic (pipeline) delay. In general, exponential delays have impulse response patterns with the form of Erlang probability functions:

$$w(T) = \frac{T^{(n-1)} e^{(-Tn/D)} n^n}{(n-1)!\ D^n} \qquad (2\text{-}4)$$

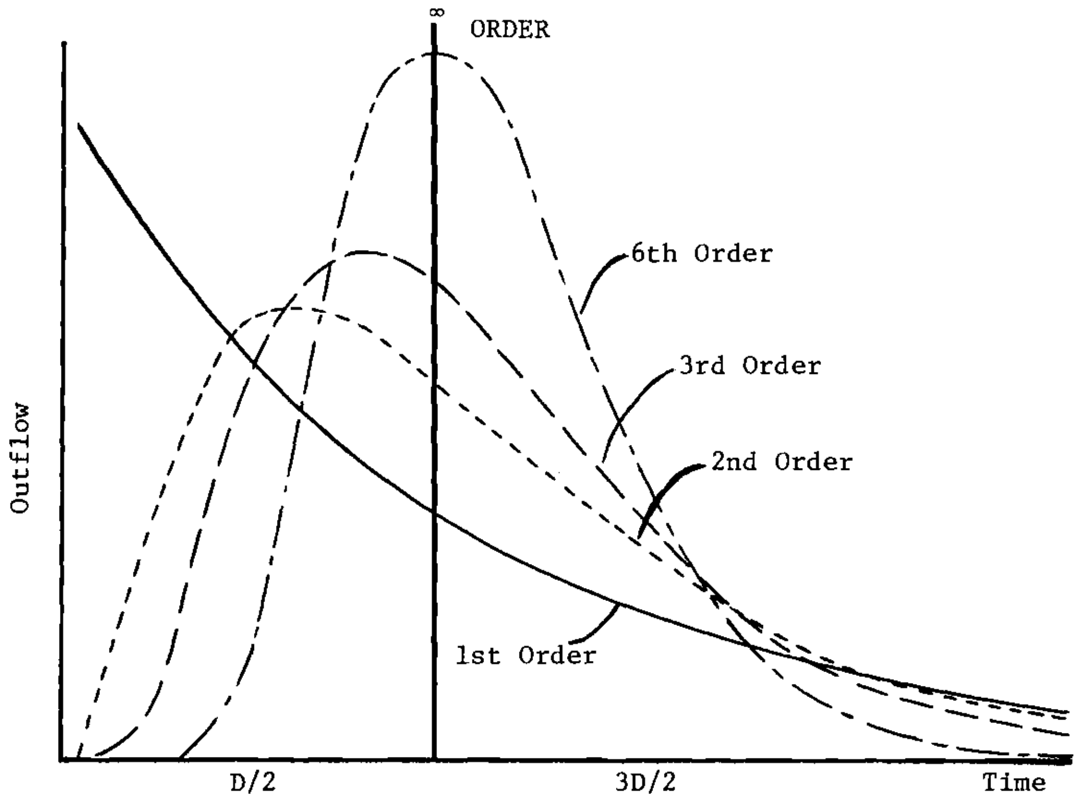


Figure 10. Response Functions for First-, Second-, Third-, and Sixth-Order Exponential Delays

where w(T) is the weight pattern (impulse response), n is the delay order, and D is the mean delay time.

The output from a continuous delay is the convolution of input stream and weighting pattern:

$$O(T) = \int_0^\infty I(T-t)\; w(t)\; dt \tag{2-5}$$

where I(t) is the input at time t.

2.2.3 Differences Between Discrete and Continuous Formulations

Discrete delays are formulated to directly mimic the operation of real world delays. The characteristics of discrete, probabilistic and conservative operation are maintained in the item by item delay. Continuous delays are conservative, but neither discrete nor probabilistic.

The continuous delay's impulse response shape looks like Erlang density functions, but the delay mechanism is <u>not</u> probabilistic. Each input pulse entering an exponential delay is broken into a deterministic train, and emitted in exactly the shape described by the delay's impulse response function. Furthermore, equation (2-5) permits fractional entities to exit the delay, thus destroying the integral nature of discrete entities.

Real world delays and continuous surrogates can share one aspect -- the average time in the delay. This is because "D", the mean lag time, is set by the analyst. The shape of the delay distribution, however, cannot always be equated. Real world delays may use <u>any</u> form of probability distribution -- uniform, normal, Erlang, etc. Continuous delays are restricted to Erlang-shaped impulse response functions. In general, continuous delays require an approximation of delay distribution shape.

These differences in operation between continuous and real world delays (or equivalently; between continuous and discrete delays) cause differences in delay behavior. Quantifying the behavioral differences is the thesis's main concern.

2.3 Relevant Literature

Two major categories of literature are surveyed for analysis of delay processes. The categories are simulation and queueing literature.

2.3.1 Survey of Simulation Literature

As far as I can determine, simulation literature lacks a systematic investigation of the differences in behavior of continuous and real world delay processes. Discrete simulation users have no need to undertake such as investigation; continuous components are not used in discrete simulation. Schemes for process generators and file structures can be found in simulation texts such as Kleijnen [1975], Fishman [1973] and Emshoff [1970]. These texts discuss the discrete and probabilistic nature of discrete simulation, but do not deal with discrete delay processes in a manner that would allow comparison of discrete versus continuous approaches.

Four currently used textbooks in the continuous simulation and systems engineering field were surveyed for treatment of delay processes: Forrester [1961], Kochenburger [1972], Schweppe [1973] and Shinners [1973]. These books promote the use of continuous model components, identified in state space form or by their input-output transfer function (or equivalently, their impulse response function). While offering analytically tractable equations for continuous delay behavior, these texts do not mention the approximation that arises when continuous models are used to represent discrete real world processes.

Such an omission is understandable since the original systems for which continuous simulation was used -- electronic and hydraulic examples -- are continuous. But more recent problems addressed by state space modeling -- such as models of social and corporate systems -- are not continuous. Little attention has been given to assess the adequacy of continuous delay approximations in non-continuous systems. For example, in the book, Industrial Dynamics, Forrester states:

> Various computational processes might be used to create a delay in a flow channel within a mathematical model. We shall consider here delay functions of only a single class - exponential delays. There is no need to exclude other kinds of functions that could be used to create delays in a flow; however, the exponential delays are simple in form, and they have adequate scope to fit our usual degree of knowledge about the actual systems to be represented.

but never returns to the subject of the adequacy of exponential state space approximations.

In a more recent paper, Schaffer [1974] says:

> Delays as Random Processes
> ...an analogy between the weighting function and a probability distribution. This analogy is not coincidental. Most continuous deterministic delay processes consist, in fact, of a large number of discrete, random events, each event making up such a small fraction of the total flow that treating the delay as continuous introduces little error.

But Schaffer offers no discussion as to how large a number of events must be present before the analogy becomes appropriate. Moreover, as accurate analogy can be made only when the referent delay contains an Erlang-shaped probability distribution -- a condition rarely present.

Some simulators try to bridge the gap between continuous and discrete simulation. Pritsker [1974] presents GASP IV as a language for both discrete and continuous simulators. He argues that models can (and possibly should) contain elements of both modeling approaches. Yet he adopts directly the exponential delays from Forrester and does not discuss the effects and limitations of the interchange of state space approximations for discrete processes.

2.3.2 Survey of Queueing Literature

Beyond the simulation literature, several entries in queueing literature appeared hopeful. A slight digression illustrating the resemblance of queueing and delay processes may be helpful.

Queueing theory examines probabilistic operations on discrete entities. Queueing systems are categorized according to the classification "X/Y/S"; X defines the arrival generation process, Y the service time distribution, and S the number of servers. Items in a queueing process are "serviced" by service units before they exit. Usually there are a finite number of servers; under many traffic intensities customers need to wait for service. The total throughput time of an entity in a queueing process is equal to waiting time plus the probabilistic service time.

A queueing process can be modeled using a delay process plus an inventory. Consider a combination of an inventory coupled to a delay process as shown in Figure 11. The inventory mimics the "waiting for service" of queueing processes. The delay mimics the service operation. A delay process may be viewed as an infinite server queueing system.

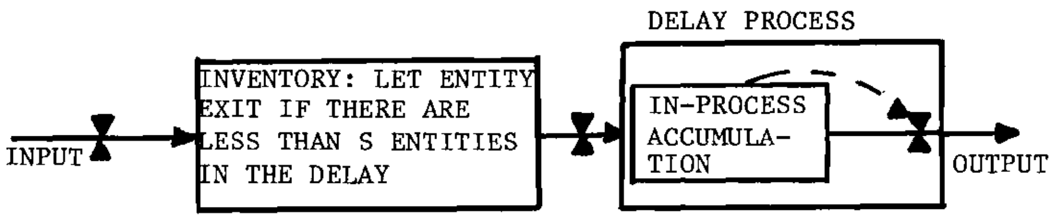


Figure 11. A Queueing Process with S Servers

The entries in the queueing literature most relevant to this thesis's problem contain derivations of the "output process" for a given queue structure. The "output process" describes the distribution and autocorrelation of interdeparture times.

Daley [1973] derives the output process of M/M/S and M/G/∞ queueing systems[1]. He states, however, "without the assumption of Poisson input, explicit computation even of the (distribution function of interdeparture times) becomes complicated." [Daley, 1973, pg 352]. Similar findings are given by Alexsandrov [1968], Daley [1968] and Mirasol [1963].

Although this literature gives insight into the operation of discrete processes, it is not helpful to the problem at hand. Output process concepts are similar to the discrete simulation literature -- directed towards next-event (e.g., interdeparture) formulations. They cannot directly be compared with the equal time interval formulation of continuous delays.

There are also two limitations on queueing literature: first, the literature describes steady state phenomena (such as steady state distributions) while the thesis addresses time-varying output properties, and second, queueing literature considers only the simplest input patterns while the thesis addresses non-Markovian input processes such as sinusoid.

2.4 Problem Delineation

The class of real world stochastic delay processes considered in this thesis is a subset of all possible delay processes. But the class considered here is general enough to describe those processes which a state space modeler would seek to approximate with a continuous delay.

---

[1] M represents Markovian (Poisson/Exponential), G represents general distributions.

2.4.1 Definition of FTU, Delay Length, and Delay Width

The thesis considers delays which have input and output pulses occurring at equally spaced time intervals. The spacing is called the fundamental time unit (FTU). Given a fundamental time unit, the probability structure for an item by item delay is a discrete distribution. The number of choices (or slots) in the distribution is termed delay "length".

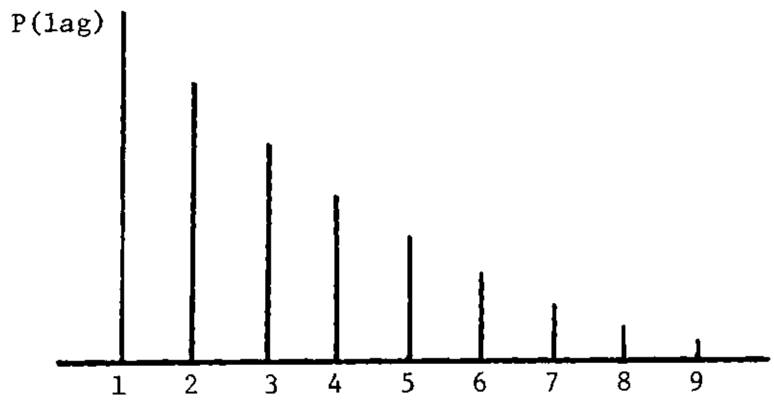


Figure 12. Discrete Exponential Distribution

Figure 12 illustrates a discrete representation of an exponential distribution with length 9. Although delay distributions are defined in a jargon of continuous functions (e.g., exponential), keep in mind that a discrete approximation is the actual distribution being used. The resolution of the approximation is determined by the value of FTU. If the FTU was halved, the exponential distribution of Figure 12 would look like Figure 13 and have a length of 18.

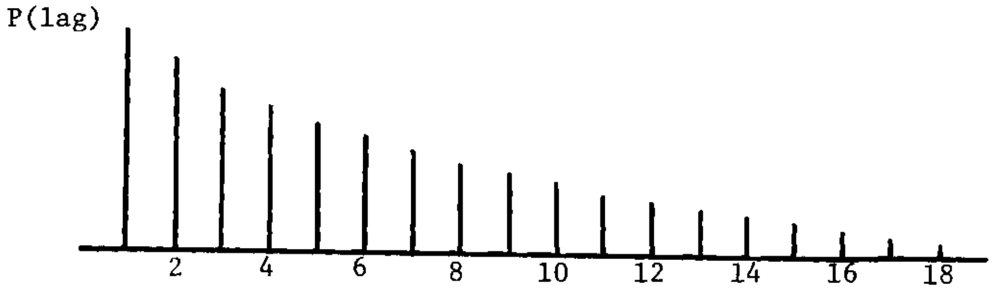


Figure 13. Redefined Discrete Exponential Distribution

Delay length is not, however, a very descriptive identifier when used in conjunction with infinite distributions, such as exponential. The length of an exponential distribution is arbitrary, depending on the accuracy desired in the discrete representation. A more descriptive measure is delay "width", a word used here to represent the range of a uniform distribution, the standard deviation of a normal distribution, and the mean of the exponential and Erlang3 distributions. Width is measured in integer number of FTUs. For example, the width of an exponential distribution of mean two weeks is two, only if FTU is one week. The width of the delay would be 14 if FTU were one day.

FTU and width (in FTUs) is set by the system; length is determined by desired resolution. I have chosen delay length to be defined to be six times the width for the normal, exponential and Erlang3 distributions, providing enough delay length to contain at least 99% of delay probability. Delay length is equal to delay width for the uniform distribution.

2.4.2 Relationships to be Addressed

The thesis derives the relationship of system parameters on the delay behavior performance measures. Performance measures include mean, variance and autocorrelation of delay output. System parameters are descriptors of the delay process and associated input series. These parameters are defined by the referent system, and are not at the liberty of the analyst. Input magnitude, frequency (for sinusoidal input), delay probability structure, and FTU are the parameters which are considered.

In order to introduce the concept of system parameters more thoroughly, let us consider a set of related processes. Each has a uniform distribution but the members differ with respect to input magnitude, delay length and FTU.

The basic system of this set has 70 entities entering a delay each FTU. Each entity of the input is stored for either one or two time units with equal probability. In steady state, the output pulses from the delay will have a mean of 70, but will deviate from this expected value due to the random selection of lag times from the delay distribution. The variation of individual output pulses is to be quantified in terms of variance and autocorrelation. Figure 14 illustrates the operation of this system.

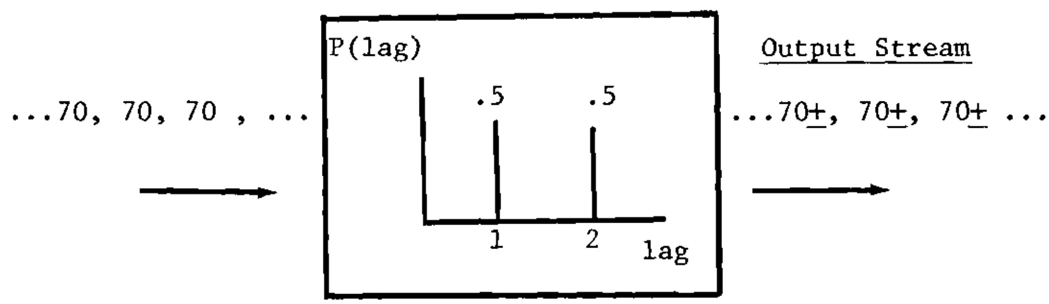


Figure 14. Basic System

2.4.2.1 Impact of Input Magnitude. Input magnitude defines the number of entities in each input pulse. In the basic system described above, the input magnitude was 70. Suppose the input magnitude of the system was altered so that only 10 entities arrive each time period. The output from this delay has a new mean (10 units) and exhibits variance and autocorrelation properties different from the basic system of Figure 14.

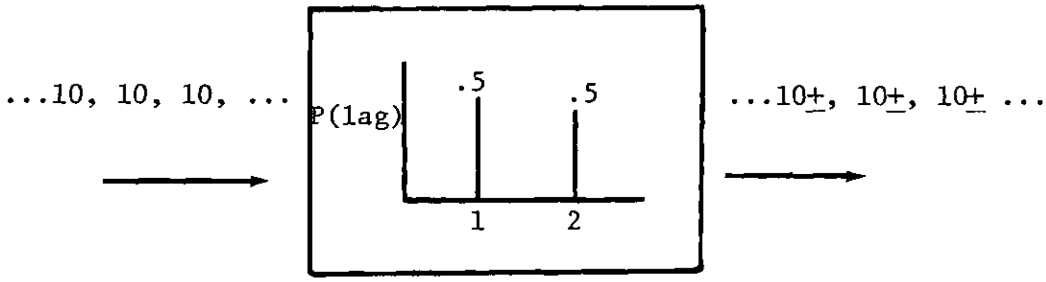


Figure 15. System with Different Input Magnitude

Question #1: What is the effect on performance when the input magnitude changes?

2.4.2.2 Impact of Delay Distribution. The delay's internal structure defining the probability of lag selection is a basic descriptor of the delay. The thesis addresses both a change in delay width, and a change in delay distribution form (exponential, Erlang3, uniform and normal distributions are presented). The figure below illustrates a change in delay width relative to the basic system. The delay distribution is uniform, but the width of the delay is now four FTUs. At each input pulse, each of the 70 entities makes a draw from a distribution containing the possibility of selecting lag times of 1, 2, 3 or 4 time units with equal probability.

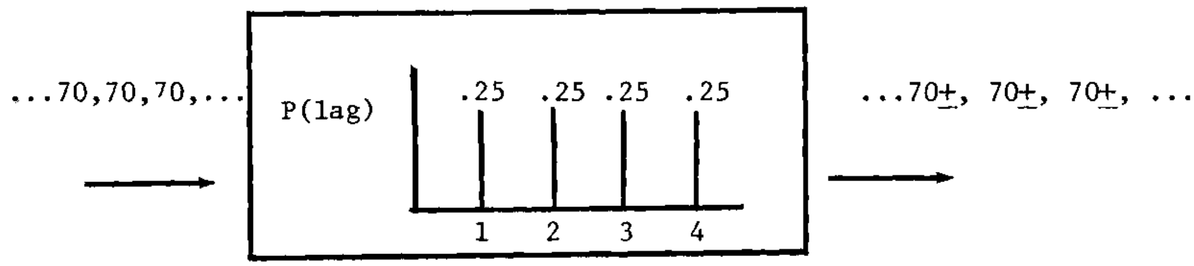


Figure 16. System with Different Delay Distribution

Question #2: What effect in performance occurs when the delay distribution is altered?

2.4.2.3 Impact of Fundamental Time Unit (FTU). In order to show how FTU may change system performance, we need to define a system in terms of absolute magnitude and absolute delay length. Consider a system with a uniform delay holding entities for a one to two week duration, and processing seventy entities weekly. Suppose a modeler received this delay description from a data collector who did not record the system's fundamental time unit. The omission of FTU leaves the delay description incomplete. Two interpretations of this description are illustrated.

The described system is the basic system of Figure 14 if the fundamental time unit is one week. All seventy entities enter the delay as a single pulse, and can draw a lag of either one or two FTUs.

Suppose, however, that the system operates with daily pulses. Then each daily input pulse consists of 10 entities. The choice of lags are 7, 8, 9, ... or 14 FTUs. The system operation is shown in Figure 17.

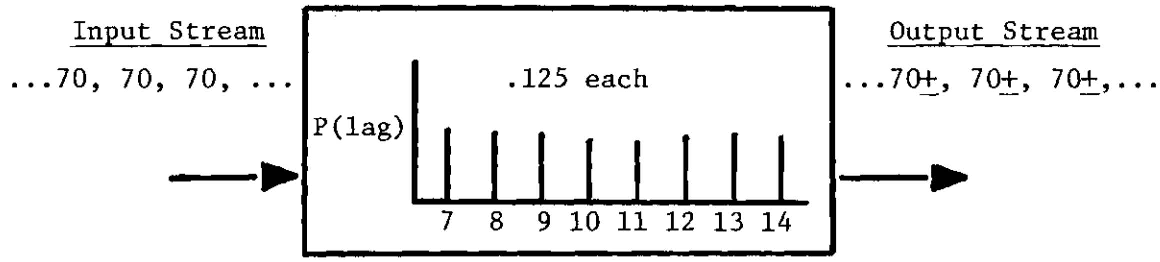


Figure 17. System with Different Fundamental Time Unit

Figure 17 illustrates that this representation of the system differs from the base case in both input magnitude and delay length. All systems have an inherent FTU. In cases of incomplete knowledge, the analyst may have the "choice" of what funadmental time unit to incorporate in the model, and needs to understand the impact of FTU on model performance. This gives rise to Question #3: What effect does FTU have on process performance?

2.5 Plan of the Thesis

The thesis is given over to answering the questions just posed. Chapter Three considers delay behavior, given constant input for three output measures: mean, variance, and autocorrelation. These output measures are related to three system parameters: magnitude, delay distribution and FTU. Results are presented for both continuous and discrete delays.

Chapter Four looks at two important cases of time varying input: single sinusoid and white noise. For the case of sinusoidal input, frequency and amplitude become system parameters. These parameters are related to the output measures: attenuation and recognition. For the case of white noise, delay behavior is measured by input-output variance gain. Parallel with Chapter Two, results are presented for both continuous and discrete delays.

Chapter Five gives preliminary consideration to the impact of delay formulation on whole model behavior. The impact on whole model behavior can be quantified only by a case by case investigation. To show how the quantitative results of Chapter Two and Three may be used to refine continuous models, a simple production-inventory problem is considered. First the behavior of the discrete referent is presented with and without noisy input. Then a series of improved continuous models are developed.

Chapter Six concludes the thesis's findings, and recommends areas of future study.

CHAPTER III

BEHAVIOR OF DELAY PROCESSES GIVEN CONSTANT INPUT

Chapter Three begins the investigation of performance differences between continuous and discrete delays. Using the modeling formulations described in Chapter Two, this chapter examines the operation of delays given constant input. The considered performance measures are mean, variance and autocorrelation.

The chapter begins (in section 3.1) by deriving continuous delay performance in all three performance measures. This presentation is straightforward due to the analytical nature of continuous delays. The derivation of equations describing discrete delays is not as easy. To introduce the idea of pulse processing in discrete delays, section 3.2 describes the processing of a single input pulse. This presentation of pulse processing is a prerequisite to the analysis of constant input processing. Representing the bulk of the chapter, section 3.3 defines constant input processing in discrete delays. Each performance measure (mean, variance, and autocorrelation) is examined individually. As a part of the examination, the impact of system parameters (input magnitude, delay distribution and FTU) on each performance measure is presented. The chapter concludes with a comparison of continuous and discrete performance given constant input.

3.1 Output From Continuous Delays

The steady state output from a continuous exponential delay is easy to determine. The output results from a convolution of the delay weighting function acting upon the input series. We have

$$O(t) = \int_0^\infty I(t-T)\, \frac{T^{n-1} e^{-Tn/D} n^n}{(n-1)!\, D^n}\, dT \qquad (3\text{-}1)$$

Here n is the order of the delay, D the mean throughput time, and I(t) the input at time t. For the case of constant input, where I(t) = N, equation (3-1) can be rewritten as

$$O(t) = N \int_0^\infty \frac{T^{n-1} e^{-Tn/D} n^n}{(n-1)!\, D^n}\, dT \qquad (3\text{-}2)$$

This equation reduces to:

$$O(t) = N,\ t \text{ large enough for steady state} \qquad (3\text{-}3)$$

The weighting function of the exponential delay is an Erlang probability function which integrates to one when t is large enough for steady state. This result is valid for all orders of exponential delays.

Since the steady state output is equal to N, the input magnitude, the analysis of system parameters on performance measures is straightforward.

$$\text{Variance } [0(t)] = E \quad [0(t) - \mu]^2 \tag{3-4}$$

Both 0(t) and μ are equal to N, thus

$$\text{Variance } [0(t) = E[(N - N)^2] \tag{3-5}$$

$$= 0 \tag{3-6}$$

$$\text{Covariance } [0(t), 0(t+i)] = E \left\{ [0(t) - \mu] * [0(t+i) - \mu] \right\} \tag{3-7}$$

$$= E[(N-N) * (N-N)] \tag{3-8}$$

$$= 0 \tag{3-9}$$

$$\text{Autocorrelation (i)} = \frac{\text{covariance } [0(t), 0(t+i)]}{\text{variance } [0(t)]} \tag{3-10}$$

$$= 0/0 \quad \text{undefined} \tag{3-11}$$

The first observation is that there is zero variance in the output series and second that the autocorrelation of the output is undefined (0/0), a reasonable result for a deterministic system.

3.2 Output from Discrete Delays with a Single Input Pulse

Before undertaking the analysis of discrete delay performance with constant input, it is helpful to see how a discrete delay transforms a single input pulse into a vector of output components. Consider the processing of a single input pulse consisting of N entities. As each entity enters the delay, an independent sample is made from the delay distribution. The delay distribution is described in terms of length "a", and P(i), i = 1, 2, ,,, "a" where P(i) is the probability of selecting a lag of "i" FTUs. If the input pulse enters at time zero, all entities must exit the delay by time "a". Defining o(i) as the output for time i, then o(i) is also the number of entities which drew a lag of t time units. The vector [o(1) ... o(a)] reports the outcome from all N independent draws from the delay distribution. The content of the o(.) vector follows a multinomial density functions, for which the following properties exist[1]:

1. the expected value of o(i) = N * P(i)
2. the variance of o(i) = N * P(i) * (1 - P(i))
3. the covariance of o(i) and o(j) = -N * P(i) * P(j)

Again, P(i) is the probability of selecting a lag of i time units. These properties are used extensively throughout the thesis.

An example is helpful: a delay structure with length 3; where lags of 1, 2, or 3 FTUs are possible with probability .2, .5, and .3 respectively. Further suppose a single input pulse enters with number of entities N = 100. The actual output may be any combination of draws from the delay distribution. Due to random selection of throughput times, the generated output series might be any of the triplets of Figure 18, or for

---

[1] See Rektorys [1969], page 1258.

that matter, any sequence that adds to 100.

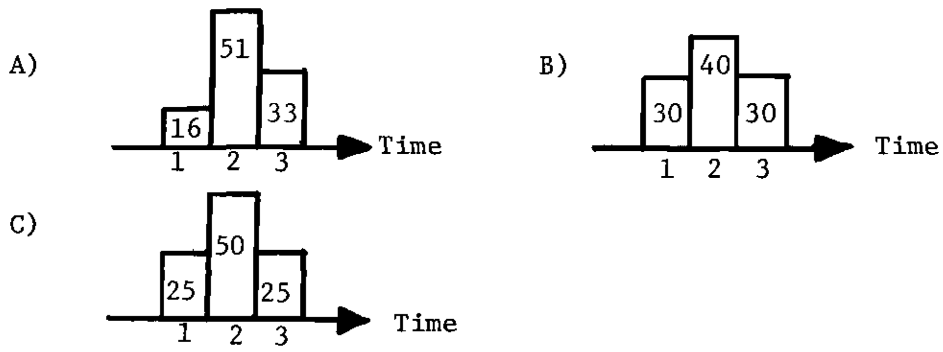


Figure 18. Possible Simulated Output

Because of the probability structure in the delay process, the values of o(.) are random variables. Using the fact that the o(.) vector follows a multinomial distribution, then:

	i=1	i=2	i=3	from multinomial property
E [o(i)]	20	50	30	N * P(i)
Var [o(i)]	16	25	21	N * P(i) * [1 - P(i)]

Furthermore, the value of o(i) is correlated with the values of $o(j \neq i)$.

Cov [o(1), o(2)] = -10

Cov [o(1), o(3)] = -6

Cov [o(2), o(3)] = -15

from

-N * P(i) * P(j)

Having established how a vector of output pulses is created by a single input pulse we now turn to output creation by a series of input pulses.

3.3 Output from Discrete Delays with Constant Input

Given constant input, closed form expressions for the delay output performance measures -- expected value, variance and autocorrelation -- can be derived by examining how output pulses are formed from a series of input pulses.

3.3.1 Conceptual Development

Section 3.2 shows how each input pulse creates a vector of output components. In constant input, these output vectors from neighboring input pulses overlap to create the total output leaving the delay. To develop this idea, consider the system of Figure 19 consisting of three identical discrete delay mechanisms. Each delay has length 3, p(.) = (.2,.5,.3). The output from the three delays add to form the system output.

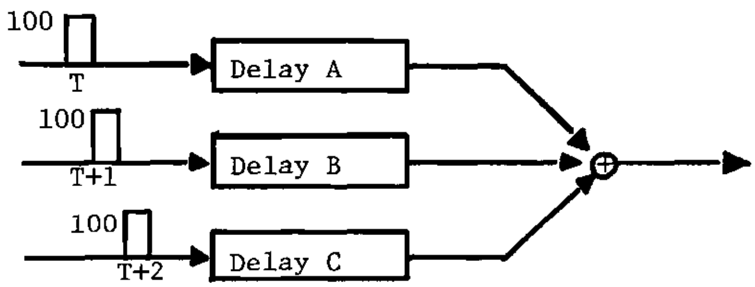


Figure 19. Multiple Delay System

Suppose that an input pulse of 100 entities enters delay A at time T. A pulse of equal magnitude enters delay B at time T+1, a third pulse enters delay C at time T+2. If we let the three sample outputs of Figure 18 represent the output from the three delays, subsystem output is as shown in Figure 20.

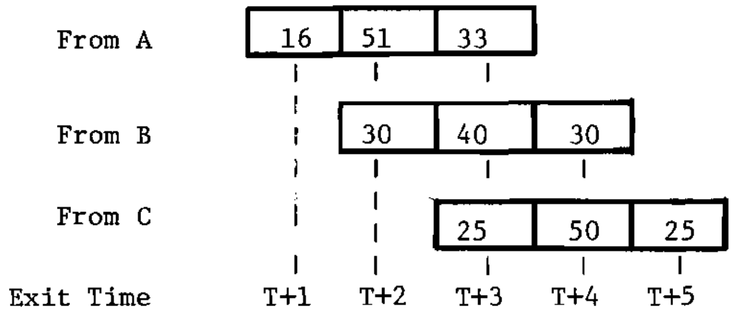


Figure 20. Possible Output from Multiple Delay System

Total outflow is 16, 81, 98, 80 and 25 for time periods T+1 through T+5 respectively.

A simpler representation of the system of Figure 19 can be created using a single delay mechanism and the property of superposition.

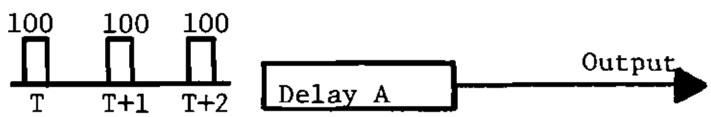


Figure 21. Single Delay System

The system portrayed in Figure 21 operates in the same fashion as the system of Figure 19. Each input pulse is independently broken into a vector of partial outputs. The individual partial outputs are stored for the appropriate duration, then emitted. The system output at time T consists of a sum of one or more partial outputs leaving at time T. If the partial outputs from the three input pulses take the form of the outputs of Figure 18, then the system output looks like:

Partial Outputs:					
from input T	16	51	33		
from input T+1		30	40	30	
from input T+2			25	50	25
Total Output:	16	81	98	80	25
Time:	T+1	T+2	T+3	T+4	T+5

Figure 22. Possible Output from Single Delay System

The generalization of this example is straightforward. Pulse "a" enters at time T and creates partial outputs a1, a2, and a3 which exit the delay at times T+1, T+2, and T+3 respectively. Pulse "b" enters at time T+1, and creates partial outputs b1, b2, and b3 which exit at times T+2 through T+4. For a string of such input pulses, the output is "built" as shown below.

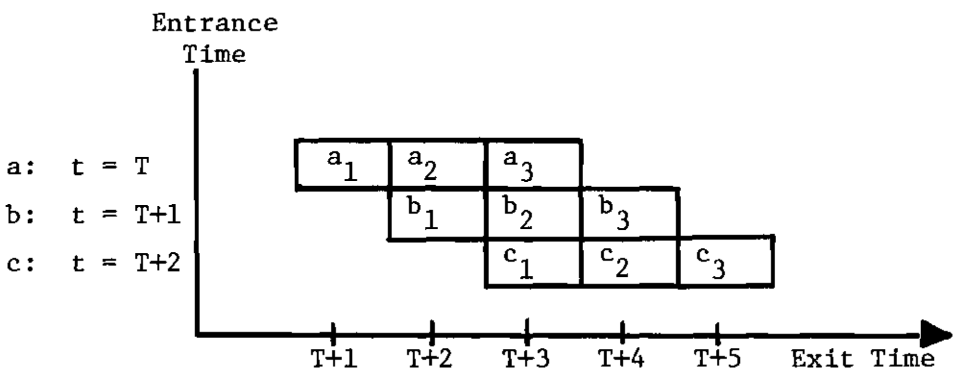


Figure 23. Building an Output Pulse

3.3.2 Expected Value of Output

The expected value of an output pulse is equal to the sum of the expected value of the pulse's components. To be specific, focus on the output at time T+3, O(T+3). (Notice that a capital "O(t)" is used to represent total output, whereas small "o(t)" denotes a partial output.) The completed output pulse consists of entities c1 which entered the delay at time T+2, and drew a lag of 1, plus entities b2 which entered at time T+1, and drew a lag of 2, plus entities a3 which entered at time T, and drew a lag of 3.

The expected value of the output pulse O(t+3) is equal to the sum of the three expected partial outputs $\overline{c1}$, $\overline{b2}$ and $\overline{a3}$. Using property #1 of multinomial distributions, the value of $\overline{c1}$, $\overline{b2}$ and $\overline{a3}$ are $N_c P(1)$, $N_b P(2)$, and $N_a P(3)$ respectively, where N_k is the number of entities entering at time k. Under the restriction of constant input, the expected value of O(T+3) is

$$E[O(T+3)] = N^*P(1) + N^*P(2) + N^*P(3) = N \sum_{i=1}^{3} P(i) = N \qquad (3\text{-}12)$$

Assuming an infinitely long series of constant input pulses, equation (3-12) reduces to

$$E[O(t)] = N \qquad (3\text{-}13)$$

This relationship is true regardless of FTU or delay distribution.

3.3.3 Variance of the Output

We now direct our attention to variance of the output series. The presentation first derives an equation defining variance, and then investigates, in detail, the impact on variance resulting from changes in the system parameters: input magnitude, delay distribution and FTU.

3.3.3.1 Development of Equations. The variance in the height of an output pulse is a function of the variance and covariance of the pulse's components. For instance, the variance of O(T+3) (in Figure 23) is equal to the sum of the variances of a3, b2 and c1, plus the sum of the covariances of (a3,b2), (a3,c1) and (b2,c1). The variance of partial outputs is defined by the second multinomial property (see Section 3.2). Independent lag draws for each time period assures zero covariances. The expected variance of O(T+3) becomes

$$\begin{aligned} \mathrm{Var}\,[O(T+3)] &= N * [P(1)(1-P(1)) + P(2)(1-P(2)) + P(3)(1-P(3))] \\ &= N * \sum_{i=1}^{3} P(i)(1-P(i)) \end{aligned} \tag{3-14}$$

For the general constant input case, the variance of output leaving a delay is

$$\mathrm{Var}[O(t)] = N * \sum_{i=1}^{a} P(i)(1-P(i)) \tag{3-15}$$

where "a" is the delay length. Note that this equation is not valid for the initial "a" periods, since the output would not yet be in steady state.

Equation (3-15) can be manipulated to provide relationships linking system parameters with output variance.

3.3.3.2 Impact on Variance from Input Magnitude. The input magnitude, N, has a direct and important relationship with output variance. Since equation (3-15) is linear in N, output variance varies proportionately to changes in N.

3.3.3.3 Impact on Variance from Delay Distribution. The shape of the delay distribution does not have great impact on output variance. The impact of delay length on output variance can be seen using a variant of equation (3-15). Define γ as

$$\gamma \triangleq \sum_{i=1}^{a} [P(i)(1-P(i))] \tag{3-16}$$

Variance can be expressed using γ:

$$\text{Var} = \gamma N \tag{3-17}$$

γ then is the ratio of steady state output variance to constant input magnitude. The ratio γ is bounded by zero and one. Why? Recast equation (3-16) as:

$$\gamma = \sum_{i=1}^{a} [P(i) - P(i)^2] \tag{3-18}$$

$$\gamma = \sum_{i=1}^{a} P(i) - \sum_{i=1}^{a} P(i)^2 \tag{3-19}$$

$$\gamma = 1 - \sum_{i=1}^{a} P(i)^2 \qquad (3\text{-}20)$$

The summation of $P(i)^2$ cannot be negative, thus γ is bounded by one. Furthermore, since all $P(i)$ are positive and their sum equal to one, the summation of $P(i)^2$ cannot be greater than one. Thus, γ is lower bounded by zero.

As the delay's width increases, the number of intervals (a) used in the representation increases. Since $\sum_{i=1}^{a} P(i) = 1$, the individual $P(i)$ decrease as the number of intervals increases. More intervals cause $\sum_{i=1}^{a} P(i)^2$ to rapidly decrease because of the squaring of smaller numbers.

The ratio γ is reported in Table 1 for uniform, normal, exponential and Erlang3 distributions (Appendix 1 contains the program used to create the table). The reported values show that γ is quite close to unity for most all distributions and widths. Gamma (and thus variance) grows as delay width grows; however, the speed of growth is distribution dependent. The normal distribution exhibits the steepest ascent, and uniform the slowest.

3.3.3.4 Impact on Variance from FTU. The fundamental time unit impacts output variance through its influence on input magnitude and delay width (which impacts γ). These two influences act in different directions, but the change in input magnitude dominates. To show this, recall the systems portrayed in Figures 14 and 17, which for convenience reappear as Figures 24 and 25 below.

Table 1. Gamma as a Function of

Distribution Type and Width

Width	Uniform	Exponential	Normal	Erlang3
1	.00	.54	.73	.53
2	.50	.76	.86	.74
3	.67	.83	.91	.82
4	.75	.88	.93	.86
5	.80	.90	.94	.89
6	.83	.92	.95	.91
7	.86	.93	.96	.92
8	.88	.94	.96	.94
9	.89	.94	.97	.94
10	.90	.95	.97	.94
12	.92	.96	.98	.95
14	.93	.96	.98	.96
16	.94	.97	.98	.96
18	.94	.97	.98	.97
20	.95	.97	.98	.97

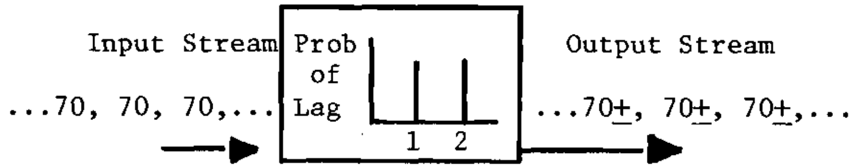


Figure 24. Basic System (Repeat of Figure 14)

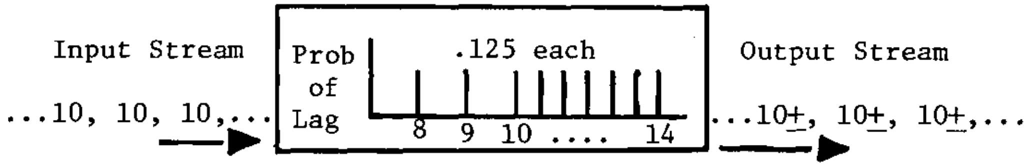


Figure 25. System with Different FTU (Repeat of Figure 17)

Consider the variance of the output from each system. Figure 24 portrays a system with input pulses of 70 entities passing through a uniform delay of width 2. The variance for such a system, using equation (3-17) is (70)(γ). In this case, γ is .50 so the expected variance is 35.

The system depicted in Figure 25 has input pulses of magnitude 10 passing through a uniform delay of width 8. The corresponding variance is (10)(0.88) or 8.8, a significant reduction. FTU impacts both the magnitude of input pulses, and γ. But since γ has an upper bound of one (and is usually close to one for most delay distributions), the change in γ, as FTU decreases, is smaller than the corresponding change in input magnitude.

The systems of Figures 24 and 25 differ only in fundamental time unit; they share the same input magnitude for an absolute time interval, 70 per week, and the same lag distribution -- uniform over two weeks or fourteen days. If Figure 24 represents a base case with variance 35,

Figure 25 displays a related system with seven times the resolution -- its variance is 8.8.

The problem at hand is how to generalize the FTU change, or change in resolution, influence on output variance. The scheme adopted here tabulates results for four distributional patterns -- uniform, normal, exponential, and Erlang3. For any distribution, resolution is noted as the number of FTUs in the distribution's absolute width.

A specific delay structure will have some constant number of entities entering each FTU. Replace N by N^*, the input pulse magnitude when FTU is set equal to the absolute width of the delay. Exception: for uniform delay distributions, N^* is defined as the number of entities per pulse when FTU is one half of the delay width[1].

Output variance is a function of FTU as given by

$$\text{Variance (FTU)} = (\gamma^*)\ (N^*) \qquad (3\text{-}21)$$

γ^* is the ratio of output variance to input magnitude for a related base case system, thus reflecting the influence of FTU change on both system width and input magnitude. Table 2 -- which is derived as a modification of Table 1 -- presents γ^* (FTU) values for the uniform, normal, exponential and Erlang3 distributions.

To use equation (3-21) we must first calculate N^*, the input magnitude for the appropriate base case, then look up γ^*. Consider the example of changing FTU to switch from Figure 24's system into that of

[1] A one FTU wide uniform distribution is deterministic.

Table 2. γ^*: Variance/N^*

D[1]	Uniform[+]	Normal	Exponential	Erlang3
1	0	.729	.540	.530
2	.500	.430	.375	.370
3	.335	.302	.277	.273
4	.250	.233	.218	.215
5	.200	.189	.180	.178
6	.167	.159	.153	.152
7	.143	.137	.133	.131
8	.126	.121	.118	.116
9	.111	.107	.104	.104
10	.100	.097	.095	.094
12	.084	.081	.080	.079
15	.066	.065	.064	.064
20	.050	.050	.049	.049
M	1/M	1/M	1/M	1/M

[+] Uniform: base case where width is 2 FTUs

[1] Number of FTUs in delay distribution width

Figure 25. Seventy units are arriving each FTU when the delay width, "D", is equal to 2. Recalling the special definition for N^* for uniform delays, we see that N^* is 70. If FTU is changed such that the delay width, "D" becomes 8, Table 2 indicates that the output variance is (.126)(70) or 8.8 (as previously calculated). If FTU decreases further to make the delay width 12 FTUs, then the expected variance is .084 times 70 or 5.8.

Another example may be helpful. Consider a system with a normal lag structure of width (standard deviation) equal to 3 FTUs and input magnitude 25 units. What would the variance be if FTU decreased by 50%? For this problem N^* is (3)(25) or 75 units. Output variance when the width is 3 FTUs is (.302)(75) or 22.65. To decrease FTU by 50% implies a delay width of 6 and output variance .159 times 75 or 11.93.

The tabulation of the four distributions in Table 2 indicates that an increase in FTU increases output variance (and vice-versa); and further, that for all distributions γ^* is approximately 1/M.

3.3.3.5 Simultaneous Consideration of Systems Parameters. The results presented thus far have treated output variance as a function of input magnitude, delay distribution width, and FTU. To show the interaction between these model parameters, an equational form for variance containing all three factors is developed.

To begin, we note that the values for γ closely follow the rule:

$$\gamma \simeq \hat{\gamma} = 1 - \int_0^\infty P^2(x)\, dx \qquad (3\text{-}22)$$

where P(x) is the probability density function. This equation is the continuous parallel to the discrete equation (3-20). The approximation is made by letting FTU become infinitely small and the number of slots (delay length) become infinitely large.

$\hat{\gamma}$ can be easily found for the cases of uniform and exponential delays. When P(x) is uniform over the interval (a,b), $\hat{\gamma}$ is

$$\hat{\gamma} = 1 - \int_a^b \frac{1}{(b-a)^2}\, dx = 1 - \frac{1}{(b-a)} \tag{3-23}$$

For instance, a delay of length 2 (containing the choices of 1 or 2 FTUs) is modeled as covering the (a,b) interval of (.5, 2.5). Given this restriction on a and b, the distance between a and b is equal to L, the length of the delay distribution. Since variance is γN, we estimate variance as $\hat{\gamma} N$ or

$$\text{Variance}_{\text{Uniform}} = N - \frac{N}{L} \tag{3-24}$$

For a family of related delay structures, both N and L are functions of the definition of FTU. Define N^* and L^* to be the values of N and L when FTU equals one. Then we have

$$N = (N^*)\ (\text{FTU})$$

$$L = (L^*)/(\text{FTU}) \tag{3-25}$$

and $$\text{Variance}_{\text{Uniform}} = (N^*)(\text{FTU}) - \frac{(N^*)(\text{FTU})^2}{L^*}$$

FTU must be large enough so that one entity (at least) enters which implies that FTU must be greater than $1/(N^*)$. Further, L^* must be greater than one FTU.

For the case of an exponential delay distribution, $\hat{\gamma}$ is given by

$$\hat{\gamma} = 1 - \int_0^{\infty} \left(\frac{1}{\mu}\right)^2 e^{-2/\mu T} \, dT = 1 - \frac{1}{2\mu} \tag{3-26}$$

Thus variance has the form

$$\text{Variance}_{\text{Exponential}} \cong (N^*)(\text{FTU}) - \frac{(N^*)(\text{FTU})^2}{2\mu^*} \tag{3-27}$$

Here μ^* is defined to be the mean throughput time when FTU is equal to one.

The Erlang3 γ values are very close to γ values for the exponential distribution. Thus the same variance equation may be used.

For the normal delay, γ when width = σ is nearly the same as the value for γ in the exponential structure, but with width = 2μ. This suggests variance for the normal delay may be approximated as

$$\text{Variance}_{\text{Normal}} = (N^*)(\text{FTU}) - \frac{(N^*)(\text{FTU})^2}{4\sigma^*} \tag{3-28}$$

Here σ^* is defined to be the distribution width when FTU is one.

3.3.4 Output Autocorrelation in Discrete Delays

Having derived the first and second moments (i.e., mean and variance) of the output series, it is natural to turn the investigation to the temporal structure of variance. The temporal structure is autocorrelation, which indicates the relationship existing between the height of neighboring output pulses. After the development of necessary equations, this section shows how autocorrelation is dependent upon delay distribution and FTU, but not upon input magnitude.

3.3.4.1 Development of Equations. A closed form expression for discrete delay autocorrelation can be derived, starting from the equation for autocovariance:

$$\text{Autocov (lag i)} = E[(O(t)-\mu)(O(t+i)-\mu)] \tag{3-29}$$

where μ is the mean of the output series. We estimate μ by N and perform the multiplication with the expectation to produce equation (3-30).

$$\text{Autocov (lag i)} = E[O(t)O(t+i) - N*O(t) - N*O(t+i) + N^2] \tag{3-30}$$

Using the fact that the expected value of a sum is equal to the sum of the expected values, equation (3-30) becomes:

$$\text{Autocov (lag i)} = E[O(t)O(t+i)] - N*E[O(t)] - N*E[O(t+i)] + N^2 \tag{3-31}$$

The expected value of both O(t) and O(t+i) is N, and equation (3-31) collapses to:

$$\text{Autocov (lag i)} = E[O(t)O(t+i)] - N^2 \qquad (3\text{-}32)$$

For simplified notation, let Y equal O(t), and Z equal O(t+i). The major task now is to determine the value of E[Y*Z]. The value of this expectation can be determined if Y and Z are replaced by their components. Figure 26 illustrates the components of output pulses, as given in the example of Section 3.3.1.

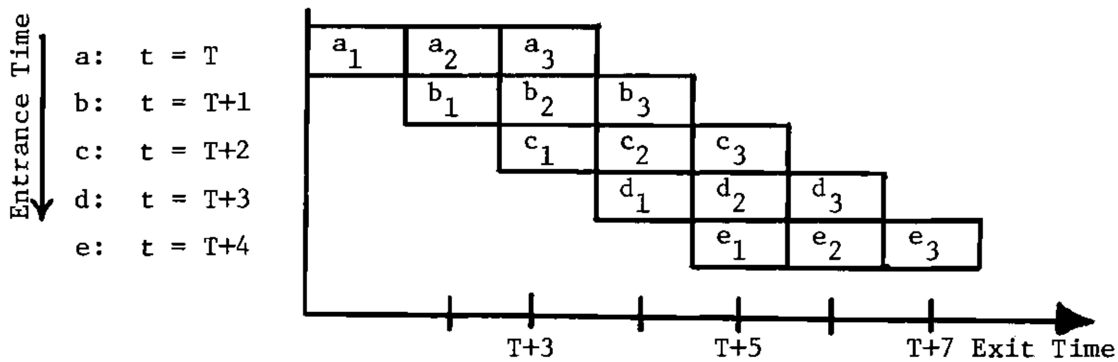


Figure 26. Creation of Output Pulses

In order to calculate lag one autocovariance, let Y = O(T+3) and Z = O(T+4). Thus we have:

$$E(Y*Z) = E[(a3 + b2 + c1) * (b3 + c2 + d1)] \qquad (3\text{-}33)$$

which is equivalent to:

$$\begin{aligned} E(Y*Z) = E(&a3b3 + a3c2 + a3d1 + \\ &b2b3 + b2c2 + b2d1 + \\ &c1b3 + c1c2 + c1d1) \end{aligned} \qquad (3\text{-}34)$$

Since expectation of a sum is equal to the sum of expected components:

$$E(Y*Z) = E(a3,b3) + + \ldots + E(c1,d1) \qquad (3\text{-}35)$$

The expectation of each product is equal to the product of the expected values of each variable plus the covariance of the variables. For example, E(a3,b3) = E(a3) * E(b3) + Cov(a3,b3). Expanding all terms, redefining expected values in terms of P(i)*N and compressing terms yields:

$$E(Y*Z) = [P(1) + P(2) + P(3)]^2 N^2 + Cov(a3,b3) + Cov(c1,d1) \qquad (3\text{-}36)$$

The sum of P(1) + P(2) + P(3) is one. All covariances operating on variables with different alphabetic designators are zero because of the independent draws given to units from different input pulses. Only the Cov(b2,b3) and Cov(c1,c2) are non-zero. Thus we have:

$$E(Y*Z) = N^2 + Cov(c1,c2) + Cov(b2,b3) \qquad (3\text{-}37)$$

Using this value for E(Y*Z), and substituting into equation (3-32), the lag one autocovariance becomes:

$$\text{Autocov (lag 1)} = Cov(c1,c2) + Cov(b2,b3) \qquad (3\text{-}38)$$

The value of the right hand covariances are -N*P(i) P(j) where P(i) and

P(j) are the probability of selecting a draw from interval i and j. In the example, Cov(c1,c2) is -N (.2)(.5) and Cov(b2,b3) is -N (.5)(.3).

The lag 2 autocovariances are computed in a similar fashion, using Y = 0(T+3) and Z = 0(T+5), and the result is that only the Cov(c1,c3) is non-zero, hence,

$$\text{Autocov (lag 2)} = \text{Cov(c1,c3)} \qquad (3\text{-}39)$$

In this example, the autocovariances for lags larger than two units are zero since the output pulses do not share components of the same input pulse.

A general form for the autocovariances of the output coming from a distribution of length a is presented in equation (3-40).

$$\text{Autocov(lag i)} = -N \sum_{j=1}^{a-i} [P(j)\ P(j+i)] \quad \text{for } 0 < i < a \qquad (3\text{-}40)$$

$$\text{Autocov(lag i)} = 0 \quad \text{for } i \geq a$$

In the above equations "i" denotes the separation between pulses.

Autocorrelations are found by dividing the autocovariances by the expected variance, which is computed by equation (3-15). The general equation for autocorrelation is presented in equation (3-41).

$$\text{Autocorrelation (lag i)} = \frac{\sum_{j=1}^{a-i} P(j)\,P(j+i)}{1 - \sum_{i=1}^{a} P(i)^2} \quad \text{for } i < a \qquad (3\text{-}41)$$

$$\text{Autocorrelation (lag i)} = 0 \quad \text{for } i \geq a$$

The autocorrelation functions for uniform, exponential, normal and Erlang3 delays are presented as Tables 3 through 6.

3.3.4.2 Impact of System Parameters on Autocorrelation. The impact of system parameters on autocorrelation can easily be investigated. A change in input magnitude N does not affect the autocorrelation values since N appears in both the variance and covariance equations, and cancels when autocorrelation is formed.

A change in the delay width does influence autocorrelation. An increase in width causes a reduction in the maximum absolute value of autocorrelation values. However, the autocorrelation values are non-zero for more lags. This tendency toward small autocorrelation values indicates independence of an output pulse with regard to the value of its neighboring phases. This result makes sense since greater delay width means that there are more lag choices available to be selected, and thus any correlation is spread over more lag choices.

Since autocorrelation is independent of N, a change in FTU impacts autocorrelation only through the alteration of delay width. Thus an increase in FTU, which causes a smaller delay width, increases autocorrelation magnitude.

Table 3. Autocorrelations for Uniform Delays

	LAG:									
Length	1	2	3	4	5	6	7	8	9	10
2	-.50	.00								
3	-.33	-.17	.00							
4	-.25	-.17	-.08	.00						
5	-.20	-.15	-.10	-.05	.00					
6	-.17	-.13	-.10	-.07	-.03	.00				
7	-.14	-.12	-.10	-.07	-.05	-.02	.00			
8	-.13	-.11	-.09	-.07	-.05	-.04	-.02	.00		
9	-.11	-.10	-.08	-.07	-.06	-.04	-.03	-.01	.00	
10	-.10	-.09	-.08	-.07	-.06	-.04	-.03	-.02	-.01	-.00
11	-.09	-.08	-.07	-.06	-.05	-.05	-.04	-.03	-.02	-.01
12	-.08	-.08	-.07	-.06	-.05	-.05	-.04	-.03	-.02	-.02
13	-.08	-.07	-.06	-.06	-.05	-.04	-.04	-.03	-.03	-.02
14	-.07	-.07	-.06	-.05	-.05	-.04	-.04	-.03	-.03	-.02
15	-.07	-.06	-.06	-.05	-.05	-.04	-.04	-.03	-.03	-.02
16	-.06	-.06	-.05	-.05	-.05	-.04	-.04	-.03	-.03	-.02
17	-.06	-.06	-.05	-.05	-.04	-.04	-.04	-.03	-.03	-.03
18	-.06	-.05	-.05	-.05	-.04	-.04	-.04	-.03	-.03	-.03
19	-.05	-.05	-.05	-.04	-.04	-.04	-.04	-.03	-.03	-.03
20	-.05	-.05	-.04	-.04	-.04	-.04	-.03	-.03	-.03	-.03

Table 4. Autocorrelations for Exponential Delays

MEAN	LAG: 1	2	3	4	5	6	7	8	9	10
1	-.32	-.12	-.04	-.02	-.01	.00				
2	-.20	-.12	-.07	-.04	-.03	-.02	-.01	-.01	.00	
3	-.14	-.10	-.07	-.05	-.04	-.03	-.02	-.01	-.01	-.01
4	-.11	-.09	-.07	-.05	-.04	-.03	-.02	-.02	-.02	-.01
5	-.09	-.07	-.06	-.05	-.04	-.03	-.03	-.02	-.02	-.02
6	-.08	-.07	-.06	-.05	-.04	-.03	-.03	-.02	-.02	-.02
7	-.07	-.06	-.05	-.04	-.04	-.03	-.03	-.02	-.02	-.02
8	-.06	-.05	-.05	-.04	-.04	-.03	-.03	-.02	-.02	-.02
9	-.05	-.05	-.04	-.04	-.03	-.03	-.03	-.02	-.02	-.02
10	-.05	-.04	-.04	-.04	-.03	-.03	-.03	-.02	-.02	-.02
11	-.04	-.04	-.04	-.03	-.03	-.03	-.03	-.02	-.02	-.02
12	-.04	-.04	-.03	-.03	-.03	-.03	-.02	-.02	-.02	-.02
13	-.04	-.03	-.03	-.03	-.03	-.03	-.02	-.02	-.02	-.02
14	-.03	-.03	-.03	-.03	-.03	-.02	-.02	-.02	-.02	-.02
15	-.03	-.03	-.03	-.03	-.02	-.02	-.02	-.02	-.02	-.02
16	-.03	-.03	-.03	0.-3	-.02	-.02	-.02	-.02	-.02	-.02
17	-.03	-.03	-.03	-.02	-.02	-.02	-.02	-.02	-.02	-.02
18	-.03	-.03	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02
19	-.03	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02
20	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02

Table 5. Autocorrelations for Normal Delays

ST.DV.	LAG: 1	2	3	4	5	6	7	8	9	10
1	-.30	-.15	-.05	-.01	-.00					
2	-.15	-.13	-.09	-.06	-.04	-.02	-.01	-.00		
3	-.10	-.09	-.08	-.07	-.05	-.04	-.03	-.02	-.01	-.01
4	-.07	-.07	-.07	-.06	-.05	-.04	-.04	-.03	-.02	-.02
5	-.06	-.06	-.05	-.05	-.05	-.04	-.04	-.03	-.03	-.02
6	-.05	-.05	-.05	-.04	-.04	-.04	-.04	-.03	-.03	-.02
7	-.04	-.04	-.04	-.04	-.04	-.03	-.03	-.03	-.03	-.03
8	-.04	-.04	-.04	-.03	-.03	-.03	-.03	-.03	-.03	-.02
9	-.03	-.03	-.03	-.03	-.03	-.03	-.03	-.03	-.03	-.02
10	-.03	-.03	-.03	-.03	-.03	-.03	-.03	-.02	-.02	-.02
11	-.03	-.03	-.03	-.02	-.02	-.02	-.02	-.02	-.02	-.02
12	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02
13	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02
14	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02
15	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02
16	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02
17	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02
18	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.02	-.01	-.01
19	-.02	-.02	-.01	-.01	-.01	-.01	-.01	-.01	-.01	-.01
20	-.01	-.01	-.01	-.01	-.01	-.01	-.01	-.01	-.01	-.01

Table 6. Autocorrelations for Erlang3 Delays

	LAG:									
MEAN	1	2	3	4	5	6	7	8	9	10
1	-.43	-.06	-.01	-.00						
2	-.28	-.14	-.06	-.02	-.01	-.00				
3	-.19	-.14	-.08	-.04	-.02	-.01	-.01	-.00		
4	-.15	-.12	-.09	-.06	-.04	-.02	-.01	-.01	-.00	
5	-.12	-.10	-.08	-.06	-.04	-.03	-.02	-.01	-.01	-.01
6	-.10	-.09	-.07	-.06	-.05	-.04	-.03	-.02	-.01	-.01
7	-.08	-.08	-.07	-.06	-.05	-.04	-.03	-.02	-.02	-.01
8	-.07	-.07	-.06	-.05	-.05	-.04	-.03	-.03	-.02	-.02
9	-.07	-.06	-.06	-.05	-.05	-.04	-.03	-.03	-.02	-.02
10	-.06	-.06	-.05	-.05	-.04	-.04	-.03	-.03	-.02	-.02
11	-.05	-.05	-.05	-.04	-.04	-.04	-.03	-.03	-.03	-.02
12	-.05	-.05	-.04	-.04	-.04	-.04	-.03	-.03	-.03	-.02
13	-.04	-.04	-.04	-.04	-.04	-.03	-.03	-.03	-.03	-.02
14	-.04	-.04	-.04	-.04	-.04	-.03	-.03	-.03	-.03	-.02
15	-.04	-.04	-.04	-.04	-.03	-.03	-.03	-.03	-.02	-.02
16	-.04	-.04	-.03	-.03	-.03	-.03	-.03	-.03	-.02	-.02
17	-.03	-.03	-.03	-.03	-.03	-.03	-.03	-.03	-.02	-.02
18	-.03	-.03	-.03	-.03	-.03	-.03	-.03	-.02	-.02	-.02
19	-.03	-.03	-.03	-.03	-.03	-.03	-.03	-.02	-.02	-.02
20	-.03	-.03	-.03	-.03	-.03	-.03	-.02	-.02	-.02	-.02

3.4 Differences Between Behavior of Discrete and Continuous Delays

The primary qualitative difference in performance of discrete and continuous formulations is the "noisiness" in the output from the discrete version. Both formulations' output series have expected value equal to the input mean. The output series of a discrete delay contains stochastic variation for which we have quantified variance and autocorrelation. The magnitude of variance depends on input magnitude, FTU and delay distribution. The autocorrelation structure depends on FTU and delay distribution but not on input magnitude.

It is beneficial to see limiting cases where both approaches yield similar results. Two obvious limiting cases present themselves: a very large number of entities per pulse, or a very small FTU. If we let the number of entities per pulse, "N", grow to ∞, equation (3-15) shows that the output variance of the item by item delay also grows to ∞. This result is unexpected since continuous flows are often visualized as an infinite number of infinitely small entities. We expect "smoother" outflow resulting from an increased input magnitude. This counterintuitive finding is resolved by noting that the coefficient of variation is a better estimate of smoothness than is variance. The coefficient of variation decreases as N increases, and in the limit, the coefficient of variation goes to zero as N goes to ∞.

If we investigate the second possibly limiting case, interesting results are found. As FTU goes to zero, several simultaneous changes occur. γ grows to one due to the redefinition of delay width in the metric of the new FTU; the input magnitude correspondingly is reduced.

The net result is that the output variance reduces as seen in Table 2.

Because the input stream cannot be fractional, the reduction of FTU cannot go far enough to permit the output variance to fall to zero. The smallest non-zero value of input is one entity. If FTU is reduced further, the input ceases to be constant (i.e., it becomes a series of ones and zeros). Thus the output variance, given constant input, is lower bounded by $\gamma^*(1)$, or just γ.

We have derived behavioral characteristics for delays receiving constant input. We now turn to Chapter Four which presents results for time-varying input.

CHAPTER IV

BEHAVIOR OF DELAY PROCESSES GIVEN TIME-VARYING INPUT

This chapter investigates delay behavior in the face of time-varying input. This is a logical extension of the analysis presented in Chapter Three where input is non-time varying, i.e., constant. The intent of the presentation is to derive the differences between continuous and discrete delays experiencing time-varying input. To limit the discussion, only two forms of time-varying input are considered: sinusoid and white noise.

The chapter is presented in four sections. The first section provides a conceptual understanding of the source of output variance. It demonstrates that output variance can be decomposed into two terms: one descriptive of the input series, and the other descriptive of internal delay randomness (which is present in discrete delays but not continuous delays). Sections 4.2 and 4.3 addresses characteristics of sinusoid processing. The attenuation of sinusoid amplitude is considered in section 4.2. The analysis includes the assessment of impact from system parameters: input magnitude, delay distribution FTU and frequency. Section 4.3 investigates recognition of the fundamental input frequency as it appears in the output series leaving a delay. Recognition is a problem in the discrete formulation due to the probabilistic nature of discrete delays. The final section derives equations describing the processing of white noise input. The primary performance measure is the ratio of output variance divided by input variance. As in all of the above sections, the presentation separately discusses discrete and continuous delay formulations.

4.1 Decomposition of Output Variance Given General Input

Chapter 3 shows that continuous and discrete delays differ in performance. The output of discrete delays exhibits stochastic variation not found in continuous delays due to the probabilistic internal structure of discrete delay mechanisms. Here we demonstrate that for general stationary input, a delay's output variance may be partitioned into two pieces -- that due to the delay's internal distribution and that due to the input variation. The presentation first addresses output from discrete delays and later addresses output from continuous delays.

4.1.1 Components of Output Variance from Discrete Delays

The variance of a stationary output series is by definition a function of the deviation of the individual output pulses, $O(t)$, from the output mean μ. The deviation of individual output pulses can be expressed, however, as the sum of two quantities:

$$O(t) - \mu = \underbrace{[O(t) - \overline{O}(t)]}_{\substack{\text{discrepancy due} \\ \text{to internal} \\ \text{delay randomness}}} + \underbrace{[\overline{O}(t) - \mu]}_{\substack{\text{discrepancy due to} \\ \text{structure of input}}} \qquad (4\text{-}1)$$

where:

$O(t)$ is the actual delay process output at time t.

μ is the output mean. This is estimated by N for stationary input.

$\overline{O}(t)$ is the expected output at time t, given the known input series up to time t.

$\overline{O}(t)$ is a linear combination of past input values. The convolution summation which defines the form of linear combination is a function of the delay's lag distribution, P(.). For example, a delay may lag entities 1, 2 or 3 FTUs with probability .2, .5 and .3 respectively. The expected output $\overline{O}(t)$ is $.2I(t-1) + .5I(t-2) + .3I(t-3)$, $I(t)$ being the input at time t.

In equation (4-1) the first bracketed variation is due to within process draws, the second bracketed variation due to the input values. These are two independent phenomena, and the terms $[O(t) - \overline{O}(t)]$ and $[\overline{O}(t) - N]$ are independent. Exploiting this we can partition total output variance into two parts:

$$\underset{\text{Total Variance}}{E[O(t) - N]^2} = \underset{\text{Due to Process}}{E[O(t) - \overline{O}(t)]^2} + \underset{\text{Due to Input}}{E[\overline{O}(t) - N]^2} \tag{4-2}$$

or

$$Var_{Total} = Var_{dp} + Var_{di} \tag{4-3}$$

To illustrate the separation of variance components consider an Erlang3 delay receiving an input pattern as shown in Figure 27. The input consists of a sine wave modulated over a DC carrier with N entities per time pulse.

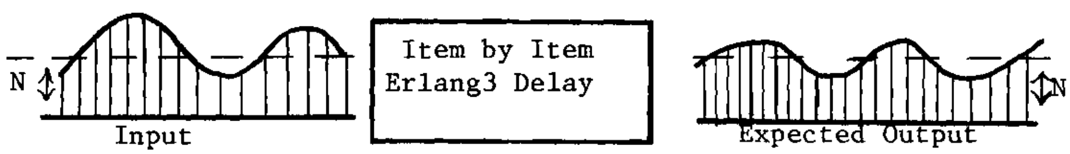


Figure 27. Expected Output of Item by Item Delay

It is reasonable to expect the output series to resemble a sinusoid pattern (we address this more explicitly in the next section of this chapter). The amplitude of the sinusoid component of the expected output series influences variance, since variance is calculated from the long term average N. Figure 28 shows the component of Var_{di} at time t_1: $[\bar{O}(t_1) - N]^2$. Var_{di} refers to input-caused variation. Consider an extreme case: suppose the input signal has zero variance, i.e., a straight line. Then Var_{di} would be zero since $\bar{O}(t)$ would be equal to N for all time t. As input irregularity grows, so Var_{di} grows.

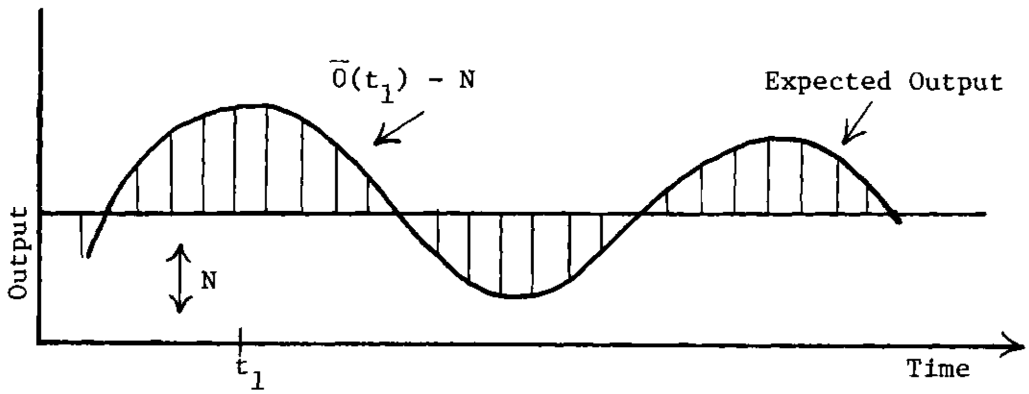


Figure 28. Components of Var_{di}

A simulated output series deviates from the expected output series in a fashion similar to the examples of Chapter 3. Figure 29 shows a simulated output series for the sinusoid input and lag structure of Figure 27. One component of Var_{dp}, that at time t_1, is illustrated, $[\bar{O}(t_1) - N]^2$ in the input series, but from the random selection of lag times. In the case of zero signal amplitude, Var_{dp} has been shown (in Chapter 3) to be a function of N (the magnitude of the carrier), and

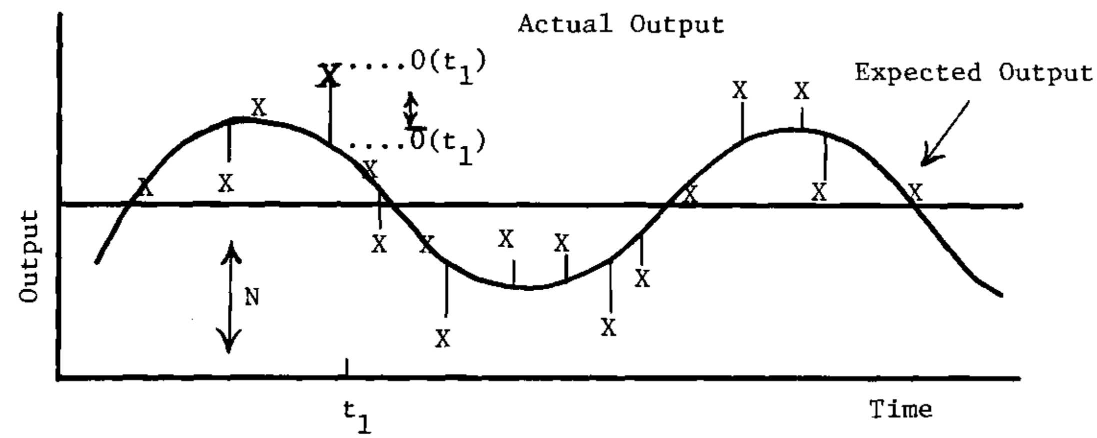


Figure 29. Components of Var_{dp}

the delay distribution (which defines γ). Given positive signal amplitude, we expect a similar variance due to internal lag selection. Note that greater deviation is expected during the peaks of the expected output series and lesser deviation during the valleys of expected output. However, the Var_{dp} component of variance when calculated over a long series of output pulses is γN.

4.1.2 Components of Output Variance from Continuous Delays

The same decomposition may be carried over to the output variance for continuous delays. For continuous delays, Var_{dp}, the variance component due to internal delay randomness, is zero. In other words:

$$O(t) = \overline{O}(t) \qquad \text{for all } t \tag{4-4}$$

Thus the total output variance for all continuous delay is created only from Var_{di}, the deviation of the expected output signal from the long run average.

$$\mathrm{Var}_{\mathrm{continuous}} = E[(\bar{O}(t) - N^2]$$ (4-5)

With this qualitative background, we now investigate the operation of delays given sinusoidal input.

4.2 Attenuation of Sinusoidal Signals

Attenuation refers to the suppression of amplitude of input sinusoids passing through a delay structure. The reduction of signal amplitude is measured in terms of the ratio of output amplitude to input amplitude.

Attenuation is important in the analysis of continuous models. This is because attenuation impacts the Var_{di} component of output variance. In some cases delays act like smoothing filters which reduce the variance (amplitude) of signals passing through them. The analysis of attenuation (and resulting similarity with smoothing filters) is first presented for continuous delays and then presented for discrete delays.

4.2.1 Attenuation in Continuous Delays

The output from a continuous delay is found from the convolution integral of equation (3-1), which is repeated below, D is the mean throughput time, n the order of the delay, and I(t) the input at time t.

$$O(t) = \int_0^\infty I(t-T) \frac{T^{n-1} e^{-Tn/D} n^n}{(n-1)!\ D^n} \, dT$$

For sinusoidal input of the form I(t) = A sin(wt) + N, it is easy to show that the steady state output from a continuous delay is also sinusoidal of the same frequency as the input signal[1]. Thus the output has the same form: B sin(wt+ϕ) + N. The output amplitude "B" is less than or equal to the input amplitude "A", and the output series may be shifted in phase "ϕ" radians relative to the input signal. Figure 30 illustrates the input-output relationship of a continuous delay.

$$\xrightarrow[\text{Input}]{A \sin(wt) + N} \boxed{\text{Continuous Delay}} \xrightarrow[\text{Output}]{B \sin(wt+\phi) + N}$$

Figure 30. Input-Output Relationship for Continuous Delay

The reduction in signal amplitude is measured in terms of attenuation, the ratio of output amplitude to input amplitude. The attenuation and phase shift properties of continuous delays are well known[2]. Figures 31 and 32 present graphical and analytical functions for attenuation and phase shift for the first, third and infinite order continuous delays.

Figure 31 illustrates that the signal attenuation is a function of the "time ratio" and the order of the continuous delay. The time ratio is formed by dividing the mean throughput duration D by the period ($2\pi/w$) or "P" of the input signal. Figure 31 shows that short period (or high frequency) signals are strongly attenuated (i.e., reduced) as

---

[1] Kreyszig [1967], page 71.

[2] Forrester [1961], page 416.

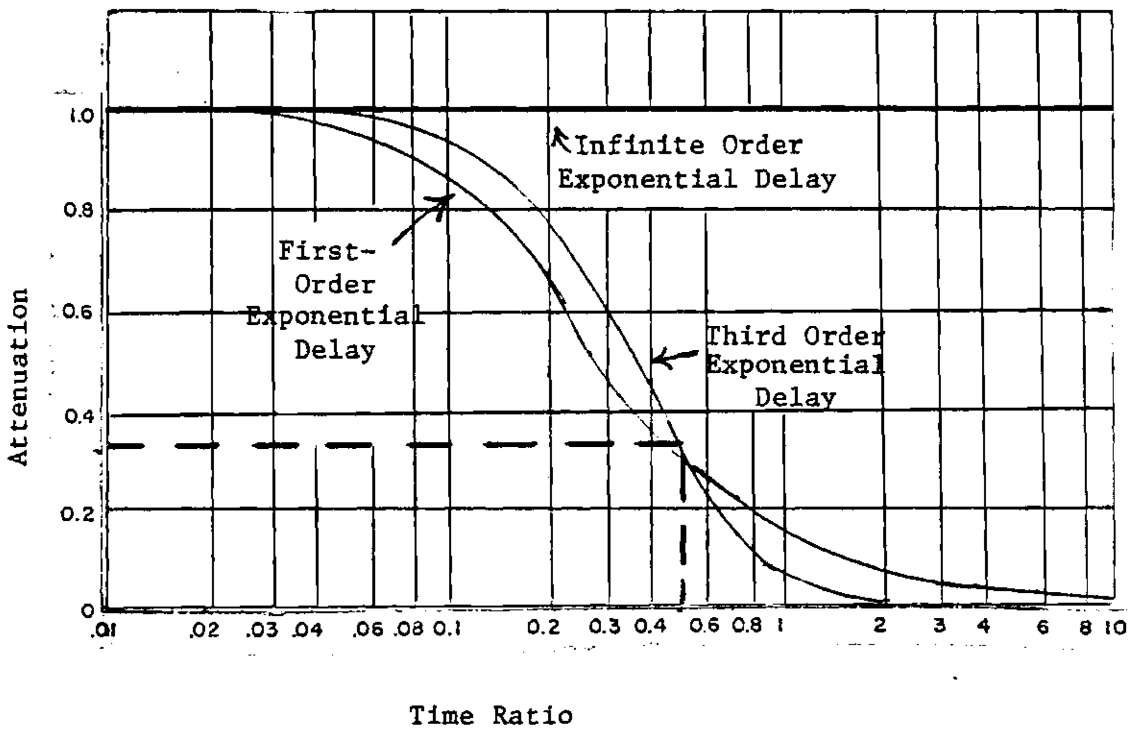


Figure 31. Attenuation versus Time Ratio

Source: Forrester [1961], page 417.

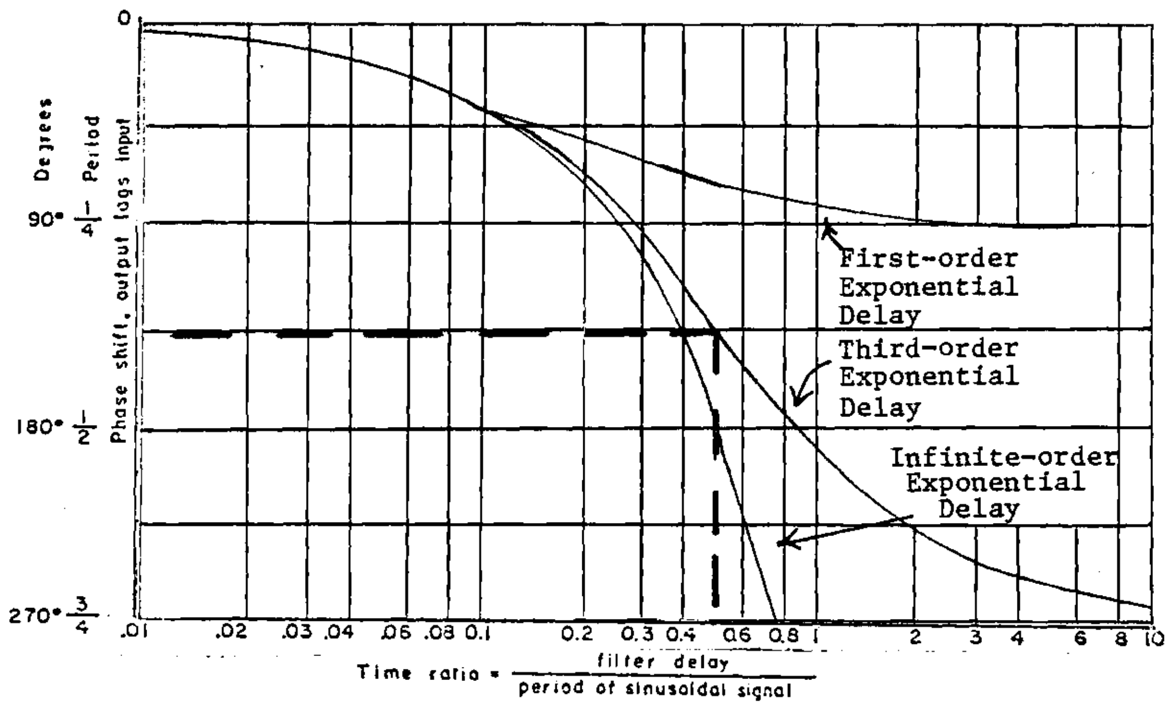


Figure 32. Phase Shift versus Time Ratio

Source: Forrester [1961], page 417.

they pass through a continuous delay. Figure 32 shows that the shift angle is also a function of the time ratio and the order of the delay.

An example: consider a sine wave of period 10 FTUs entering a third order exponential delay of mean lag 5 FTUs. Figure 31 shows that for this time ratio (5/10 = .5) the output signal retains only 33% of the input amplitude. Figure 32 shows that the output signal is shifted by 135° relative to the input signal.

Continuous delay formulations have the same structure[1] and variance reducing action as do smoothing filters. Figure 31 indicates that high frequency signals are completely suppressed, i.e., they do not appear in the output series with large amplitude (see figure below). The attenuation of high frequency signals is cited as evidence by

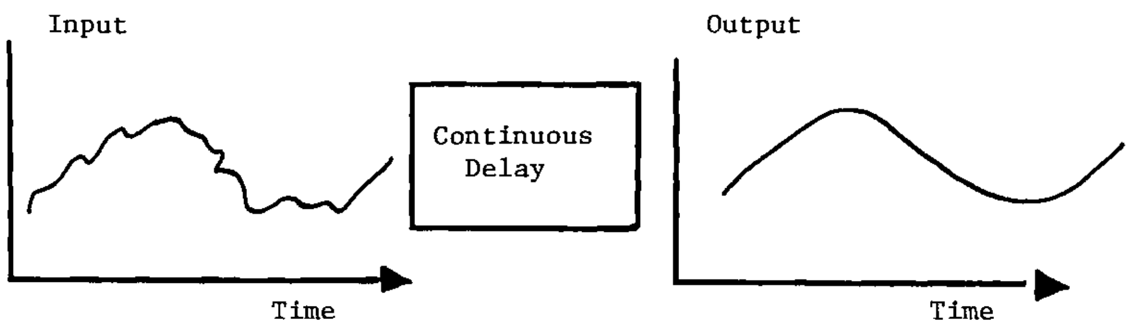


Figure 33. Smoothing of Noisy Input Signals

feedback dynamics modelers[2] who argue that their models do not call for multiple simulation runs. However, attenuation and phase shift properties of continuous delays may not adequately reflect the behavior of real world

---

1 See Wright [1976]

2 Forrester [1961], page 413.

delays.

4.2.2 Attenuation in Discrete Delays

The notion of attenuation carries over to discrete delays with two revisions in definition: we shall consider attenuation between the input series and the expected output series; and consider magnitude to mean the number of entities in the positive half wavelength of the sinusoidal signal. Justification for these definitions follows.

4.2.2.1 Conceptual Development. In discrete delays we view attenuation as the change between the input and expected output series. To see why this is done, consider the patterns shown in Figures 34 through 36. Figure 34 presents a sinusoidal signal with wavelength of 10 FTUs and magnitude 100 modulated over a carrier of 40 units.

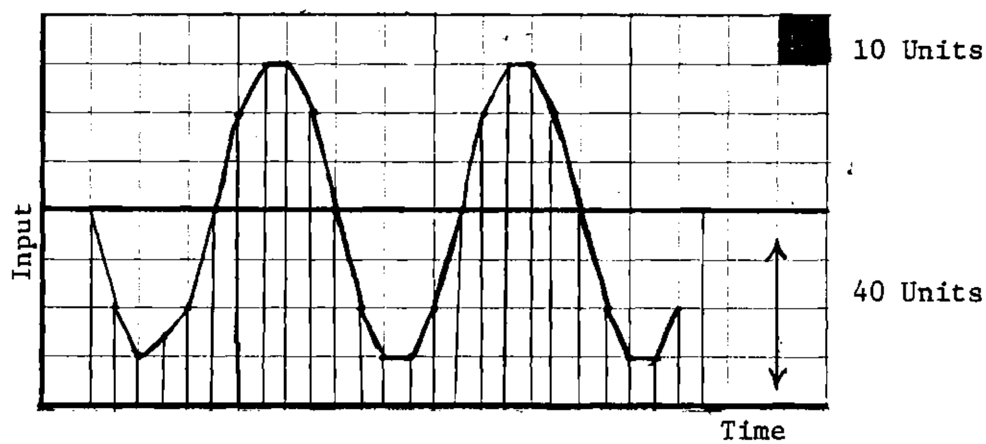


Figure 34. Discrete Sinusoidal Input

Figure 35 depicts the output from an Erlang3 delay (mean = 2 FTUs) given the input of Figure 34. The middle (heavy) line is the attenuated input signal (with magnitude 81) which is the expected output series. The outer (light) lines of the figure are the 2σ confidence limits delineating the region within which individual output pulses are most likely to occur.

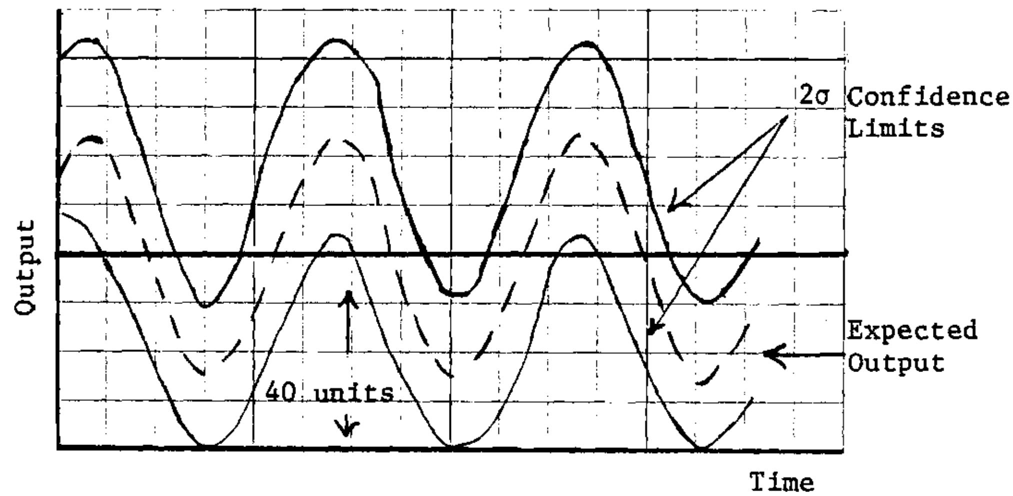


Figure 35. Expected Output and Confidence Limits

Figure 36 presents one realization for the output trajectory from the same delay mechanism and input pattern. Such a trajectory contains frequency components other than the input frequency. The noisiness of the output series (measured by Var_{dp}) may mask the input periodicity altogether.

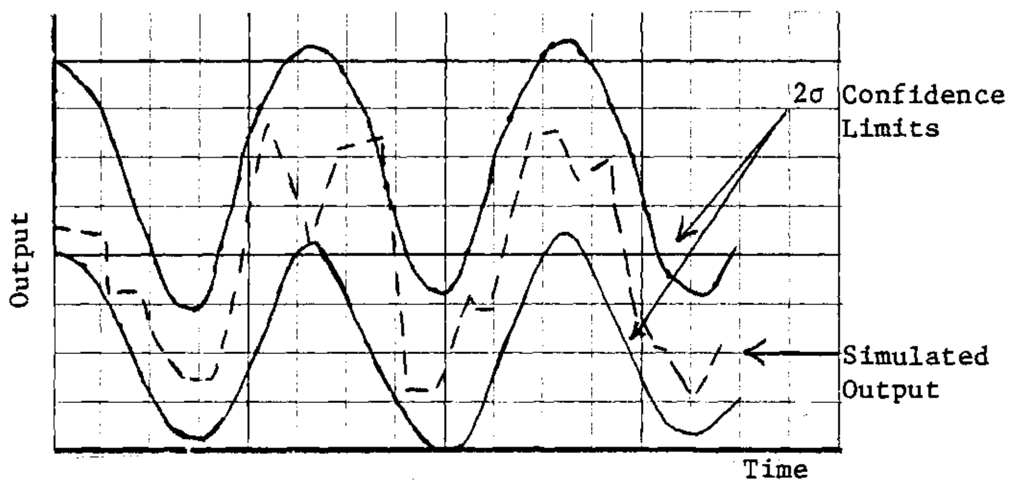


Figure 36. Simulated Output

4.2.2.2 Development of Equations. We can now develop a set of equations for the expected delay output and the variance of the output pulses given the time varying input. Consider a delay distribution of length "a" where P(i) is the probability of obtaining a lag of i FTUs. And, let I(t) be the input at time t. Then the expected value of the output at time t is

$$E[O(t)] = \sum_{i=1}^{a} [I(t-i)\ P(i)] \tag{4-6}$$

and the variance of the actual output from the above expected output is

$$\mathrm{Var}[O(t)] = \sum_{i=1}^{a} [I(t-i)\ P(i)\ (1-P(i))] \tag{4-7}$$

The derivation of these two relationships directly parallels the derivation of equation (3-15). When I(t) is constant for all t, these equations collapse to the form of equations (3-12) and (3-15).

To calculate attenuation, the input vector I(.) is constructed as a discretized sine wave:

$$I(t) = A \sin(wt) + N \qquad t = 0, 1, 2, ,,, \tag{4-8}$$

To make the signal feasible, the sinusoid is modulated over a carrier of N entities to insure that I(t) is non-negative. Using the input vector and equation (4-6) the expected output series is created.

The output signal has the same period "P" as the input signal. Thus to yield a complete wave, P consecutive output values are generated using equation (4-6). The output signal contains a phase shift ϕ.

So we have

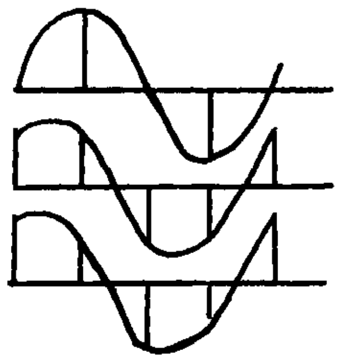
$$O(t) = B\sin(wt + \phi) + N \qquad \text{for } t = 0, 1, 2, ,,, \tag{4-9}$$

Because of the phase shift and discrete sampling of the sine wave, output amplitude is more difficult to determine than just searching to find the maximum of the output values. Why is this true? Consider three samples of O(t).

Case 1 0,100, 0,100

Case 2 70, 70,-70,-70

Case 3 86, 50,-86,-50



$\phi = 0$

$\phi = \pi/4$

$\phi = \pi/3$

Figure 37. Output Sampling

The above figure shows that the phase shift changes the output values. The three realizations above have maximum values of 100, 70, and 86 respectively, yet all three represent the same output wave. Several schemes for getting around this problem are possible; one choice is fitting a sinusoid to the P output values. A computationally simpler approach will serve. Attenuation can be equated to the ratio of output magnitude over input magnitude, both calculated over a single wavelength "WL" of the respective series (see equation (4-10)). Magnitude, then is defined as the summation of the entities of the positive half wavelength of each signal.

$$\text{Attenuation} = \frac{\sum_{t=1}^{w1} [\bar{O}(t) - N]\ u(t)}{\sum_{t=1}^{w1} [I(t) - N]\ v(t)} \tag{4-10}$$

$$u(t) = 1 \text{ if } \bar{O}(t) > N$$
$$= 0 \text{ otherwise}$$

$$v(t) = 1 \text{ if } I(t) > N$$
$$= 0 \text{ otherwise}$$

The computer routines to compute attenuation for normal and Erlang3 discrete delays are documented in Appendix 2. These distributions are of special interest to the continuous simulation community since the Erlang3 delay is often used to represent normal delays in state space modeling.

4.2.2.3 Impact of System Parameters on Attenuation in Discrete Delays. Following the continuous simulation literature, attenuation is reported as a function of the time ratio: delay width divided by input signal period, or equivalently, a function of the product: delay width times signal frequency. Four possible influences on attenuation were investigated: input signal magnitdue, FTU, frequency, and delay width.

Input signal magnitude does not influence attenuation. Any change in input magnitude is directly transferred to the output magnitude. The effect is canceled when the attenuation ratio is formed.

One might expect that the impact of FTU cancels in the same way. A change in FTU will change both the delay width and input signal period. As a unit, FTU cancels when the time ratio is formed. However, FTU does have several effects on the attenuation properties of a delay due to its impact on the resolution of the delay distribution and on input signal. A too large FTU causes the shape of the delay distribution to have little correspondence with the generic distribution (Erlang3 or normal). Of equal concern is the relationship between FTU and the shape of the input. We wish to address sinusoidal input. Higher frequency signals are poorly represented when the signal wavelength approaches the size of FTU. The highest frequency signal which can be investigated has a wavelength of (2)(FTU), and is called the Nyquist frequency[1]. However, this signal has lost most of its sinusoidal shape; it alternates in a maximum, minimum, maximum fashion.

To make the analysis valid for input of nearly sinusoidal form, and delay distribution of approximately the generic form, FTU should be less than 1/10th of the signal wavelength, and less than 1/3rd of the delay width (μ for the Erlang3, σ for the normal). The investigation reported below is based upon these constraints on FTU, and is valid only for "small" FTU systems.

Frequency and delay width directly impact attenuation as shown in the next two figures. Figure 38 illustrates expected attenuation for a discrete normal delay, Figure 39 attenuation for a discrete Erlang3 delay. In both distributions, the high frequency (or short wavelength) signals suffer great amplitude reduction. For a signal to maintain at least 50% of its amplitude, its wavelength has to be at least 5 times the standard deviation of the normal delay, or 3 times the mean of the Erlang3 delay.

---

[1] Jenkins and Watts [1968], page 53.

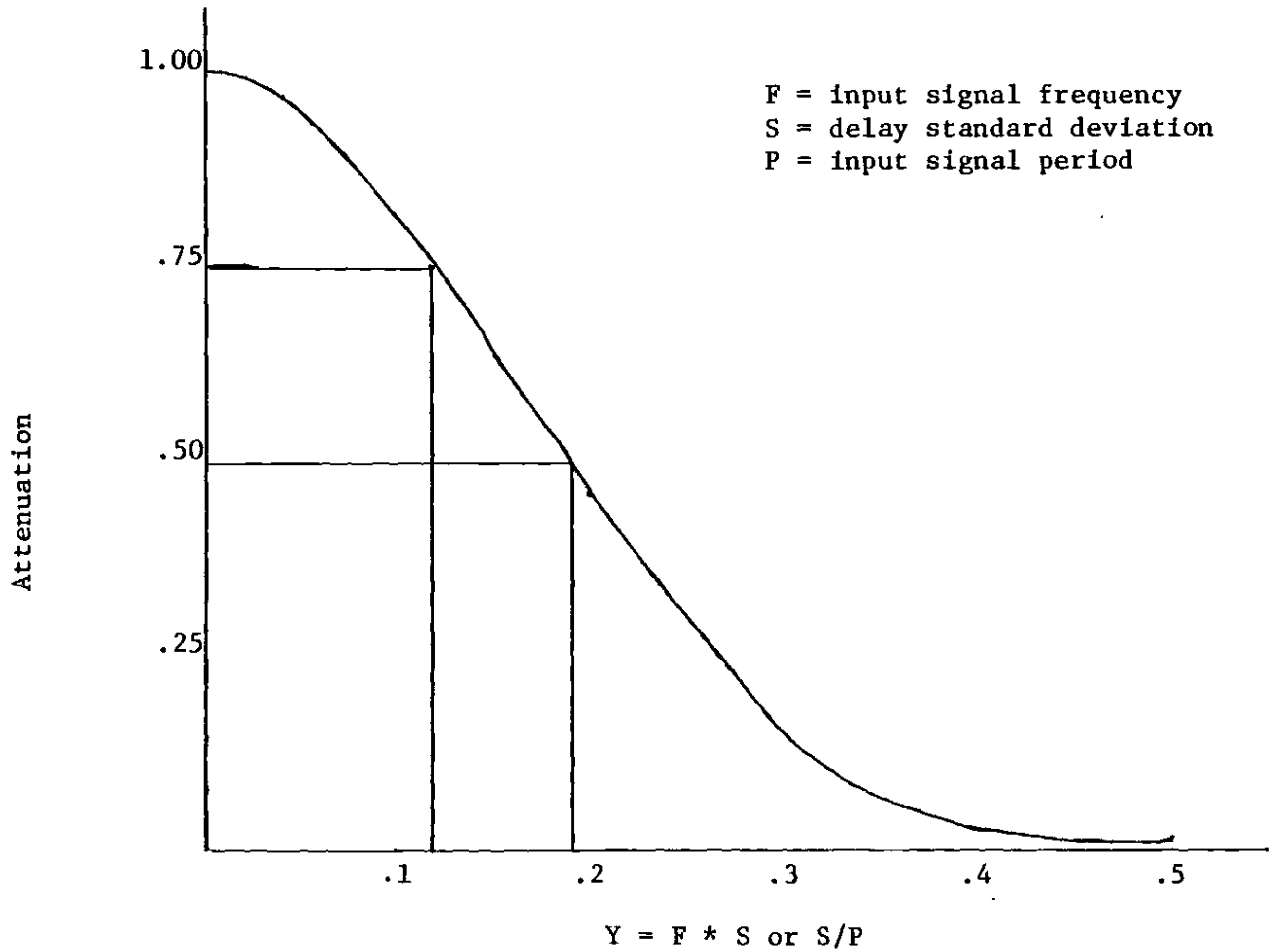


Figure 38. Attenuation for Discrete Normal Delay

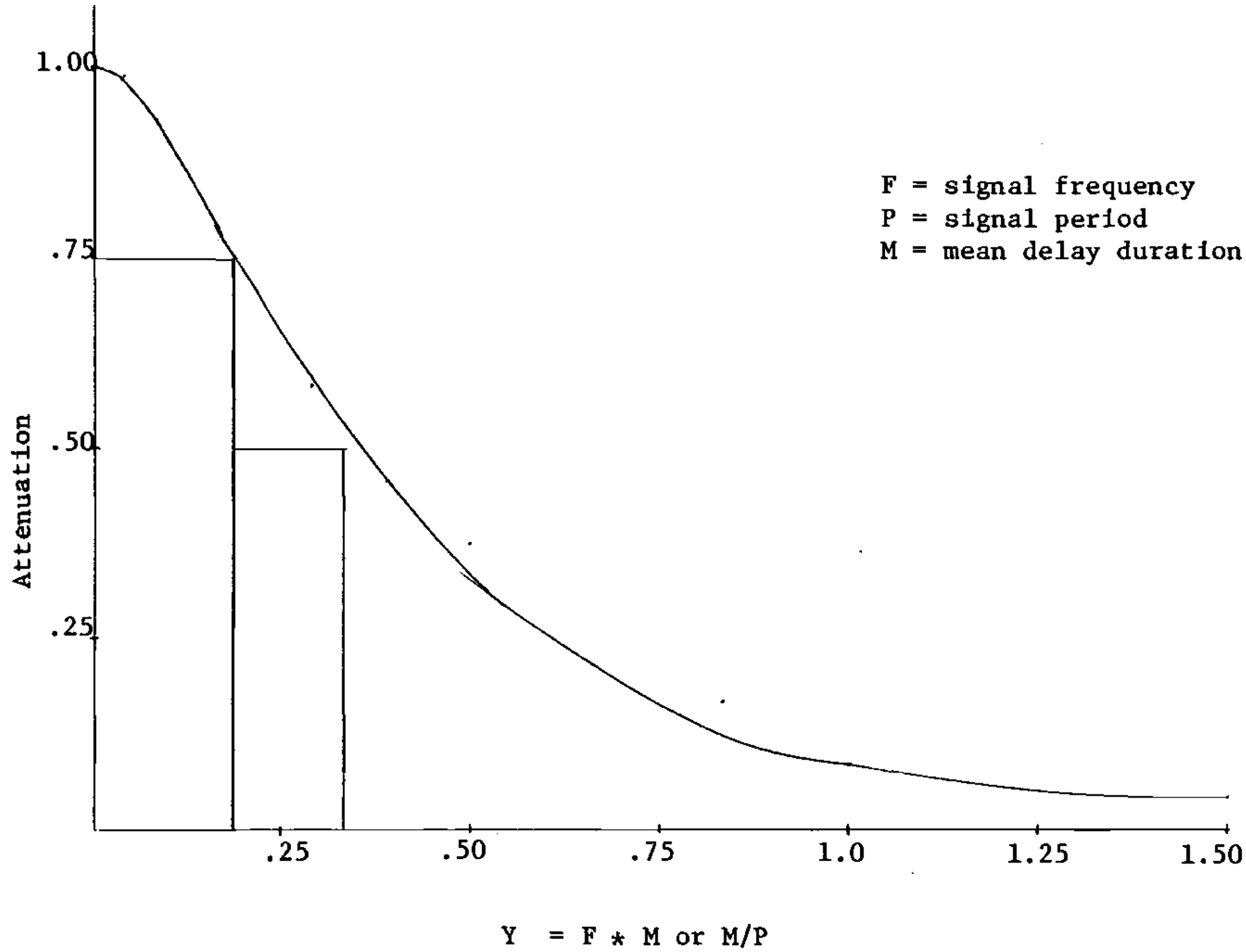


Figure 39. Attenuation for Discrete Erlang3 Delay

For an attenuation of 75%, a signal would need a wavelength 8.3 times the standard deviation of the normal, or 5 times the mean of the Erlang3 delay.

To read similar relationships off Figure 38 and 39, select an attenuation value on the vertical axis, scan across till the curve is encountered. The corresponding point on the horizontal axis defines the time ratio which has the selected attenuation property.

4.2.3 Attenuation Differences Between Continuous and Discrete Formulations

We can now describe the differences in attenuation between continuous and discrete delay formulations. Figure 39 indicates that only a small difference between the attenuation patterns of Erlang3 continuous and Erlang3 discrete delays. A comparison for normal delay formulations is not possible: there exists no way to represent the normal weighting function with a linear dynamical system.

Continuous simulators often use Erlang3 surrogates to model normal real world delays. The modeling literature[1] recommends that an Erlang3 delay can be used to model a normal delay if the mean delay times are equated. Since in steady state the content of a delay process's internal accumulations is equal to the input times mean lag time, matching of means insures similar content of the continuous Erlang3 and discrete normal formulations. However, the attenuation properties of the two delays can greatly differ. Attenuation has been shown to be a function of the standard deviation of the normal distribution; attenuation is <u>independent</u> of the mean lag of that delay.

---

[1] Wright [1976], page 12.

To illustrate how large these attenuation differences may be, consider a normal delay of mean 10 and standard deviation 2. The recommended Erlang3 delay surrogate has a mean lag of 10. A review of Figures 31 and 38 indicates that the Erlang3 mean 10 mechanism more strongly attenuates signals than the normal delay with mean 10 and standard deviation 2. For a signal of period 10, the Erlang3 shows an attenuation of .08 (time ratio M/P = 10/10 = 1). The appropriate normal delay (with time ration σ/P = 2/10 = .2) shows an attenuation of .45. This example shows equating means does not equate attenuation pattern.

An alternate scheme for approximating normal discrete delays exists; it improves the match of attenuation patterns but at a cost: steady state in-process delay contents will no longer be equal. This scheme suggests a normal (μ,σ) distribution be approximated by an Erlang3 with standard deviation equal to the normal's standard deviation.

Dispersion, rather than mean, influences the attenuation properties of a delay. Dispersion relates directly to the shape of P(.) which is used to calculate attenuation. For a uniform delay, the width of the delay is equivalent to the range of the choices of lag times. For the normal delay, it is the standard deviation that sets the width of P(.).

Why then was attenuation found to be a function of mean delay time for both the Erlang3 and exponential delays? The answer is simple. The standard deviation of the exponential is equal to the mean of the exponential, and the standard deviation of the Erlang3 distribution is related to the mean by $\sigma = \mu/\sqrt{3}$.

Figure 40 displays the graphs of 1) the Erlang3 attenuation plotted in the metric of Sigma/Period; 2) the normal attenuation plotted in the

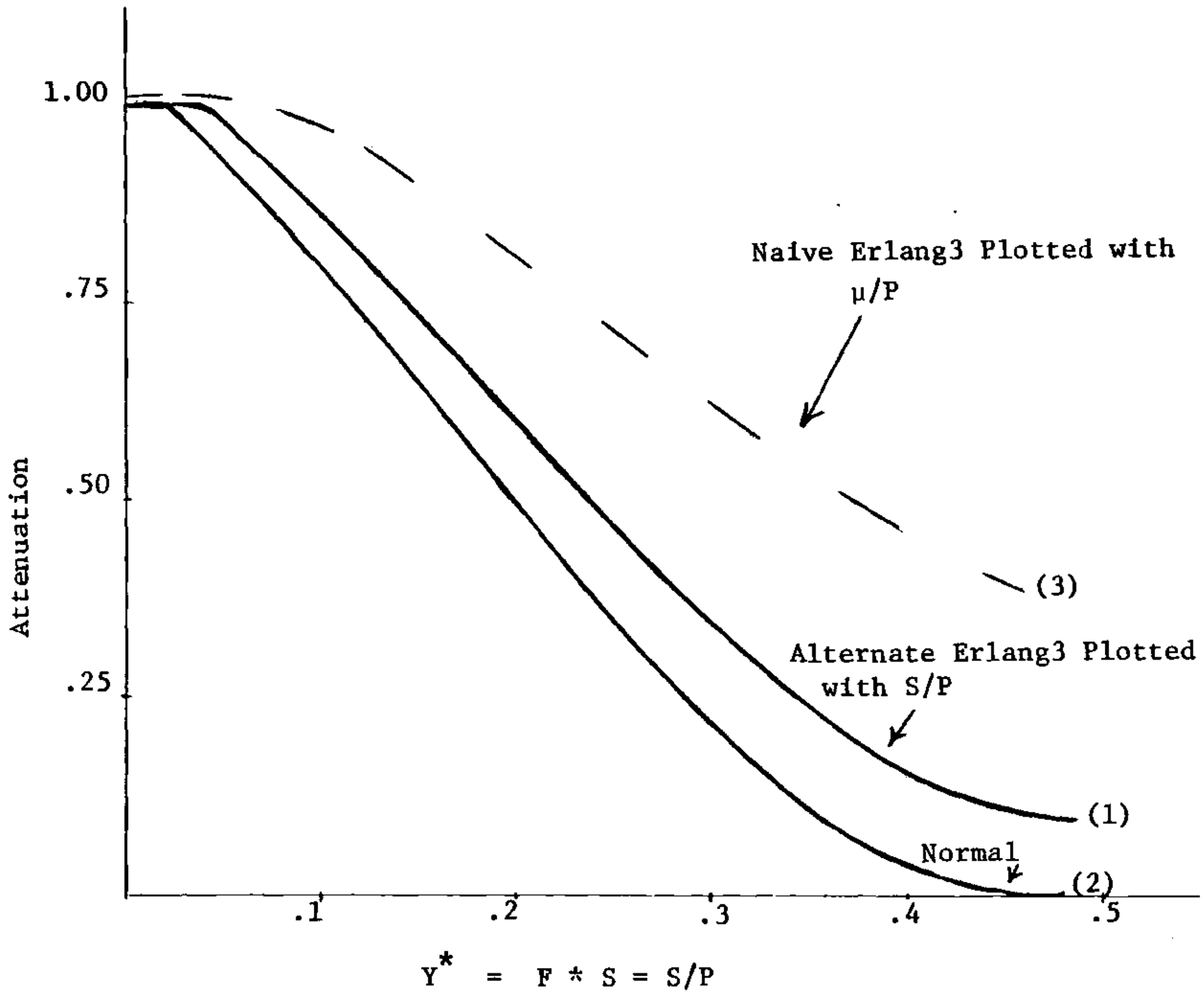


Figure 40. Alternate Erlang3 Surrogate for Normal Delays

same metric; and 3) the Erlang3 attenuation plotted in the old metric of mean/period. The Erlang3 attenuation plotted in the old metric of mean/period. The Erlang3 new-metric graph more closely resembles the graph of the normal. However, the new metric Erlang3 exhibits less smoothing of signals in the high frequency region than does the normal.

Whether the naive or alternate Erlang3 formulation ought to be used is case specific -- depending on the modelers concern for matching internal delay content or attenuation pattern. At least the need for the trade-off is now made visible.

4.3 Recognition of the Fundamental Frequency Given Sinusoidal Input

The previous section alludes to the potential disappearance of the input's fundamental frequency from the output trajectory. This problem is addressed in terms of recognition. For convenience of the development, the presentation of continuous delay recognition is postponed till after the development of recognition equations for discrete delays.

4.3.1 Recognition in Discrete Delays

Our ability to recognize a sinusoidal signal in the output series of a discrete delay depends on two factors. The first factor is the amplitude of the expected output sine wave which depends in turn on both the amplitude of the input sine wave and the attenuation imparted to a a sine wave of that frequency. The second factor constraining recognition is Var_{dp}, the variability of the output due to internal delay randomness, which is a function of the magnitude of the input carrier and the form of the delay distribution.

The analysis of the previous section considered a single frequency sine wave carried on a DC signal:

$$I(t) = A \sin(wt) + N \qquad t = 0, 1, 2, ,,, \tag{4-11}$$

A particular frequency "f" (equal to $2\pi/w$) may be easily recognized in the output series when a small carrier is used (but, of course N must be at least as large as A else the aggregate inflow becomes negative for some time periods). With a much larger input magnitude, the frequency is masked by the Var_{dp} component of output variance.

One measure of recognition is R(f): a ratio of the maximum expected output signal divided by the standard deviation of the noise produced by the carrier. The maximum expected value of the output is the product of input signal amplitude times attenuation. Attenuation is, as shown earlier, a function of input frequency. Thus R(f) is found by:

$$R(f) = \frac{\text{Attenuation}(f)\ A}{\sqrt{Var_{dp}}} \tag{4-12}$$

The variance due to internal delay randomness was shown in Chapter 3 to be equal to γN. Thus the denominator of equation (4-12) is $\sqrt{\gamma N}$. Further, since γ is nearly one for all distributions, and $\sqrt{\gamma}$ even closer to one, σ can be approximated by $\sqrt{N}$. So the recognition factor R(f) is approximated by:

$$R(f) = \frac{\text{Attenuation}(f)\ A}{\sqrt{N}} \tag{4-13}$$

R(f) must be greater than zero and since N must be at least as large as A, the recognition measure is less than Attenuation (f) * $\sqrt{A}$.

This recognition measure is reasonable. Consider Figure 41. The amplitude of sinusoid output Q falls within its carrier's one σ limit; signal Q' exceeds the one σ limit. The value of R for signal Q' exceeds the value for signal Q; and indeed Q' is more easily recognized.

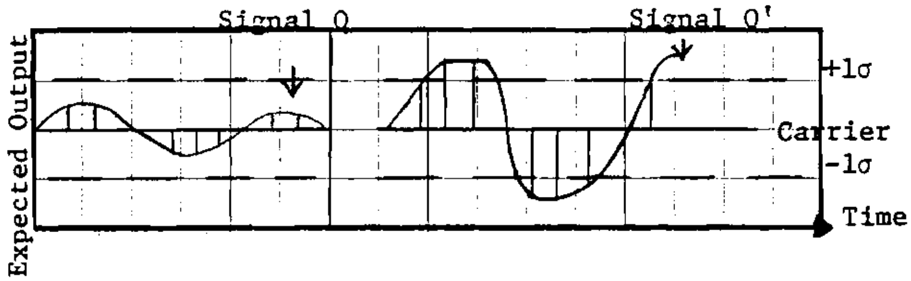


Figure 41. Recognition of Signals

With this measure we can easily show the impact of system parameters on recognition. An increase in the input amplitude "A" improves recognition. Increasing carrier magnitude "N" decreases recognition. The value of attenuation is a decreasing function of delay width. As the number of slots in a delay distribution increases, width increases, and the attenuation ratio falls. Thus wider delays reduce recognition.

The value of attenuation is also a decreasing function of frequency; higher frequency signals are more strongly suppressed and thus harder to recognize.

The last question here is how does R(f) depend on FTU? Attenuation is independent of FTU for legitimate (i.e., reasonably small compared to signal period and delay width) FTU values. Thus the impact of FTU resides in its effect on the ratio $A/\sqrt{N}$. Consider the case when FTU is halved. Both the carrier N and the signal amplitude are reduced by 50% by this operation. The ratio of $A/\sqrt{N}$ will be 70% of its original value. So as FTU decreases, the recognition potential decreases.

4.3.2 Recognition in Continuous Delays

Applying the definition of R(f), continuous delays exhibit perfect (infinite) recognition due to a zero denominator of equation (4-12). The zero results from Var_{dp} being zero. This perfect recognition of continuous delays is in contrast to the recognition of sine waves from discrete delays which is a function of the carrier magnitude, input signal frequency and amplitude and FTU.

4.4 White Noise Input

We now turn to another important case of time-varying input: white noise. White noise is an interesting special case; and one for which variance results are easily derived for both continuous and discrete delays.

4.4.1 White Noise Input in Continuous Delays

Continuous delays reduce the variance of white noise signals passing through them. To show this, we exploit a frequency domain representation of white noise. White noise can be considered as a summation of sinusoidal signals, including components of all frequencies, with all components of equal power. (Power is a function of the square of the amplitude of the signal.) A pure white noise signal is not realizable since an infinite amount of power would be required to form the infinite number of frequencies comprising the white noise signal. Band limited white noise is realizable because the component frequencies are limited to the region $(0, f_o)$.

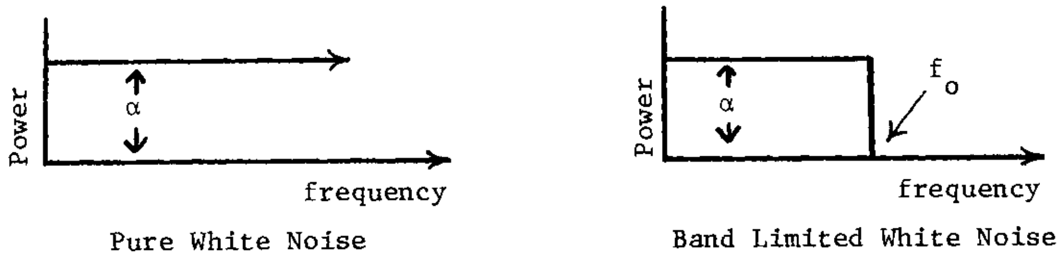


Figure 42. Pure vs Band Limited White Noise

Figure 42 illustrates the difference between pure white noise and band limited white noise. The band limited case has zero power in frequencies above f_o, power equal to α in frequencies below f_o. In sampled data systems f_o can be related to the sampling interval and the Nyquist frequency[1].

The total variance of such as input stream is determined by the integral of the power spectrum:

$$\text{Variance} = \int_{-\infty}^{\infty} P(f)\, df \tag{4-14}$$

where P(f) is the power at frequency f. The equations for the pure white noise, and band limited white noise are

---

[1] Jenkins and Watts [1968], page 53.

$$\text{Variance(pure)} = \int_0^{f_o} \alpha df = \alpha\infty = \infty \tag{4-15}$$

$$\text{Variance(limited)} = \int_0^{f_o} \alpha df = \alpha f_o = \sigma^2 \tag{4-16}$$

Relying on the property of superposition, we consider each component of the input signal independently passing through the delay. The output consists of the summation of attenuated input signals. Furthermore, the output power spectrum is[1]

$$P_{out}(f) = A^2(f) P_{in}(f) \tag{4-17}$$

A(f) represents the attenuation of the delay at frequency f. The output variance for the band limited case is therefore:

$$\text{Variance}_{\text{output}} = \int A^2(f)\alpha df \tag{4-18}$$

and has an output power spectrum of Figure 43,

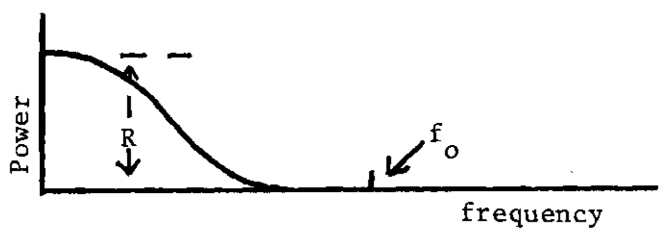


Figure 43. Power spectrum of Output

---

[1] Jenkins and Watts [1968], page 14.

resembling the square of the attenuation curve for the particular continuous delay at hand. This figure illustrates that the output signal is no longer a band limited white noise process since some component frequencies have more power than others. An implication of this "unwhiteness" is the presence of nonzero autocorrelation in the output series[1].

Another observation from the figure is that the output variance is less than the input variance. Since the attenuation function is upper-bounded by one, $P_{out}(f) \leq P_{in}(f)$ for all f. Variance is equivalent to the area under the power spectrum curve, and thus output variance is less than the input variance (see Figure 44).

Figure 44. Variance as Integral of Power Spectrum

The reduction in variance cannot be categorized in terms of variance gain. This is true because f_o, the cutoff frequency, impacts the gain ratio. If f_o is changed to twice the original value of Figure

---

[1] Values of autocorrelation can be calculated from the output power spectrum by using the inverse Fourier transform; see Jenkins and Watts [1968], page 217.

44, we have

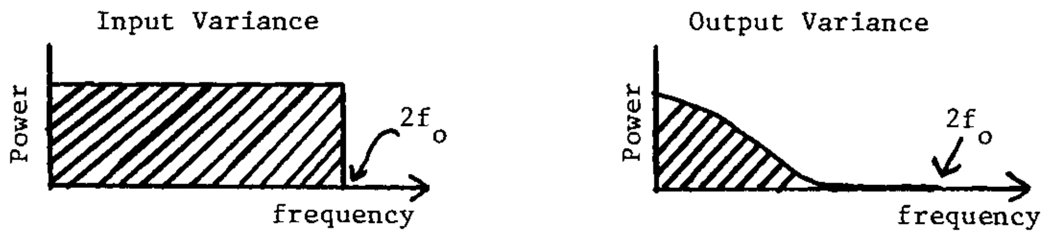


Figure 45. Variance as a Function of f_o

The increase in f_o doubles the input variance, yet the output variance remains nearly the same. Thus the variance gain ratio has been reduced by an increase in f_o.

Similar arguments show that continuous delays reduce the variance of <u>any</u> type of signal passing through it. The two fundamental equations, (4-14) and (4-17), are valid for not only white noise signals, but signals of any power spectrum composition. The argument for a reduction in variance relies only upon the condition that attenuation must be upper-bounded by one for all frequencies, a condition that is true for all continuous delays.

4.4.2 White Noise Input in Discrete Delays

We can now turn to the processing of white noise input by discrete delays. The emphasis is again on defining the ratio of output variance to input variance. The development begins with the simple delay structure used in previous examples: the three choice distribution with P(.) = (.2, .5, .3). For this delay we have already shown that:

$$\overline{O}(t) = .2\ I(t-1) + .5\ I(t-2) + .3\ I(t-3) \qquad (4\text{-}19)$$

If the input series is composed of independent pulses -- a condition true for white noise -- then:

$$\mathrm{Var}_{di} = \mathrm{Var}[\bar{O}(t)]$$

$$= .2^2\mathrm{Var}[I(t-1)]+.5^2\mathrm{Var}[I(t-2)]+.3^2\mathrm{Var}[I(t-3)] \qquad (4\text{-}20)$$

Since for a white noise input series variance for each input pulse is constant at σ^2, we have:

$$\mathrm{Var}_{di} = [(.2)^2 + (.5)^2 + (.3)^2]\sigma^2 \qquad (4\text{-}21)$$

For a general discrete delay with white noise input, equation (4-21) can be generalized into:

$$\mathrm{Var}_{di} = \sigma^2 \sum_{i=1}^{a} P^2(i) \qquad (4\text{-}22)$$

Again "a" is the length of the delay distribution, and P(i) the probability of drawing a lag of i FTU's. Recall the definition of γ from Chapter 3.

$$\gamma = 1 - \sum_{i=1}^{a} P^2(i) \qquad (3\text{-}16)$$

Substituting equation (3-16) into equation (4-22), we have

$$\mathrm{Var}_{di} = \sigma^2(1 - \gamma) \qquad (4\text{-}23)$$

Chapter 3 also derived a value for Var_{dp} (in that chapter, input variance was zero):

$$Var_{dp} = \gamma N \qquad (3\text{-}17)$$

Total variance is then:

$$Var_{Total} = Var_{dp} + Var_{di} \qquad (4\text{-}24)$$

$$= \gamma N + \sigma^2(1 - \gamma)$$

An intriguing result appears when we consider variance gain -- the ratio of output to input variance -- for a white noise signal passing through a discrete delay. Equation (4-24) may be recast as:

$$\zeta = \frac{\text{Output Variance}}{\text{Input Variance}} = \frac{Var_{Total}}{\sigma^2} = (1 - \gamma) + \frac{\gamma N}{\sigma^2} \qquad (4\text{-}25)$$

ζ is the ratio of output variance to input variance. ζ is greater than one if N/σ^2 is greater than one. In other words, when the input variance σ^2 is greater than N, ζ is less than one. When σ^2 is less than N, ζ is greater than one. Thus discrete delays tend to funnel the input variance to produce output variance which is close to N, the average input level.

Figure 46 illustrates the funneling effect for a uniform delay receiving an average of 50 entities per time unit. The abscissa defines the delay distribution range, the ordinate defines output variance. A

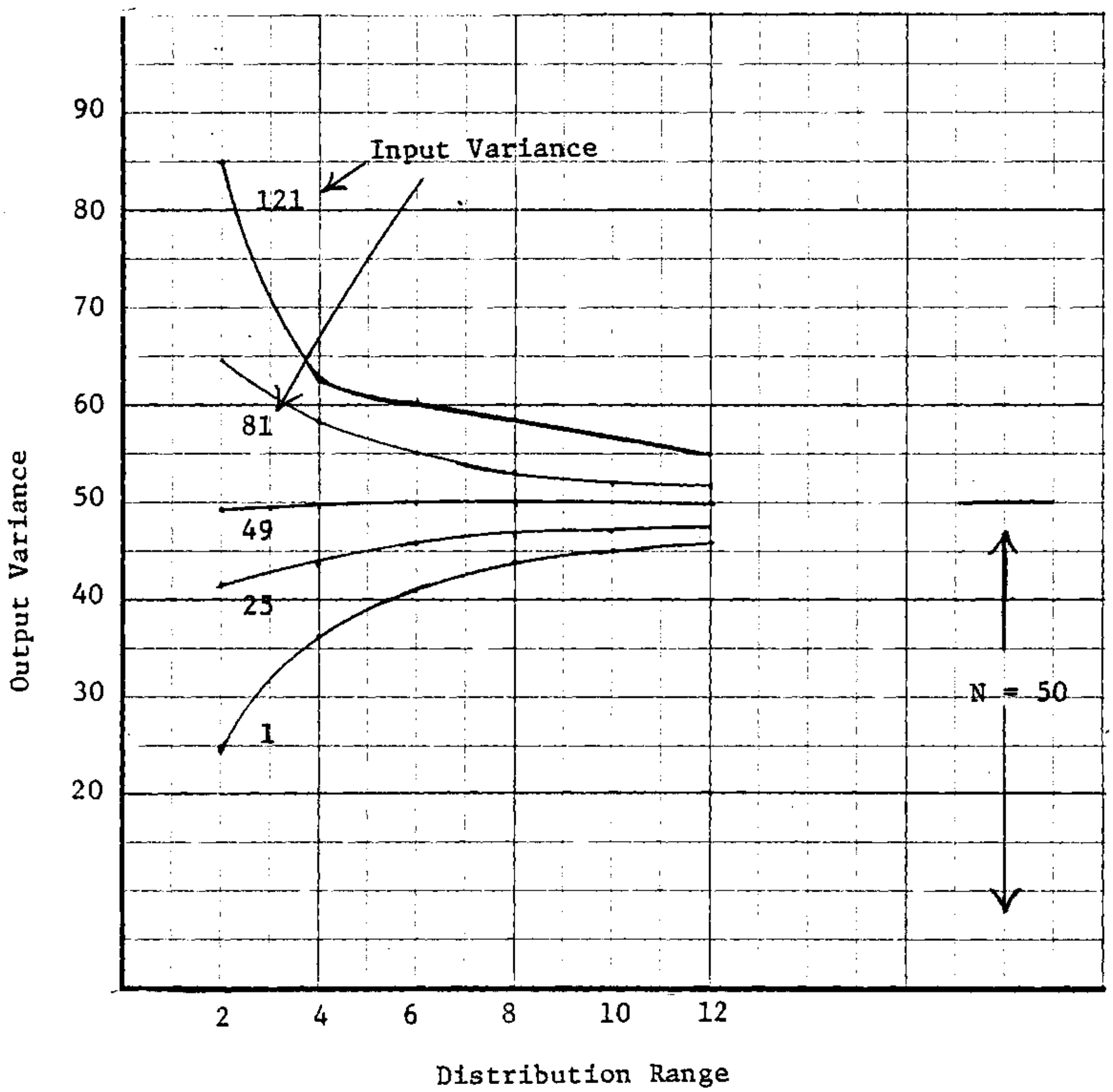


Figure 46. Funneling of Output Variance

family of curves defining input variance is presented to show how output variance is funneled towards N, which is 50 in this example.

4.4.3 White Noise Input - Difference Between Continuous and Discrete Formulations

The difference between continuous and discrete performance can be traced to the two components of variance Var_{di} and Var_{dp}. Var_{di} has been shown to be less than or equal to the input variance in both discrete and continuous formulations. Var_{di} is equal to $(1-\gamma)\sigma^2$ for the discrete case, and $1/f_o \int_0^{f_0} A^2(f)\, df\sigma^2$ for the continuous case. Differences arise when the Var_{dp} component of variance is considered. Var_{dp} is zero for continuous delays, but equal to γN for the discrete delay. This extra variance may cause the discrete formulation's output variance to exceed its input variance.

We have now completed the analysis of behavior of individual delay processes. To illustrate how the approximate behavior of idealized continuous delays affect multicomponent systems, we turn to the analysis of a complete closed loop system which contains a delay process.

CHAPTER V

BEHAVIOR OF A SIMPLE CONTROL SYSTEM WITH ALTERNATIVE DELAY FORMULATIONS

Thus far the thesis addresses the performance differences between delay formulations. The performance measures which are discussed focus on the behavior of delay processes so that behavior equivalence of two delays can be judged. (Thus accuracy can be judged when a continuous approximation to a discrete process is used.) The analysis has not considered the relationship of the delay accuracy to total model adequacy when the delay is used as a component of a larger model. This relationship is very important, but difficult to identify. In a case specific analysis qualitative indications of this relationship can be produced.

This chapter presents one such analysis. A simple inventory-production system is examined to determine the significance of delay formulation interchange. Alternate formulations are realized with a set of continuous models and a set of continuous models. Each model is exercised with the same starting conditions, and performance differences noted. This chapter concludes with a presentation of corrective techniques which can be used to reduce the difference in model performance.

5.1 Problem Description

The example inventory-production system is chosen to be quite simple, but not atypical. Here, an inventory supervisor is charged with maintaining inventory at a desired level -- 40 units. At the end of each working day, he receives a report of that day's sales and ending inventory. He uses both information sources to request new units from the production department. The request policy which he uses is

$$\text{Requests} = \frac{40 - \text{inventory}}{2} + \text{sales} \qquad (5\text{-}1)$$

Fractional valued requests are rounded up or down according to a toss of a coin.

The production department receives the requested order the next work day and immediately initiates work on the ordered units. Each unit is individually fabricated. The production time for units is known to follow an exponential distribution with a four day mean production time.

Two variants of this system are considered: in the first sales is assumed to be constant at 4 units per day; in the second sales is a draw from a truncated normal population of mean 4 and variance 3.5 (but no non-zero entries). The performance measure for both systems is the variance of day by day inventory around the desired value of forty units.

5.2 Solution Using Discrete Formulations

To provide references to judge continuous models, we develop discrete models of the example system.

5.2.1 Discrete Model Design

The discrete formulation is straightforward, precisely corresponding to real world operation. A GPSS approach to discrete modeling is employed. The moving entities for the model are the items in production. Entities are generated each day by the production request made by the inventory supervisor. Each entity selects a draw from an exponential distribution to determine how long it must remain in production. The production delay is formulated as a file structure; each element of the file represents an in-process entity marked with its exit time. Upon leaving production, entities enter an aggregate inventory from which undifferentiated sales entities are withdrawn. Entity flow is shown in Figure 47.

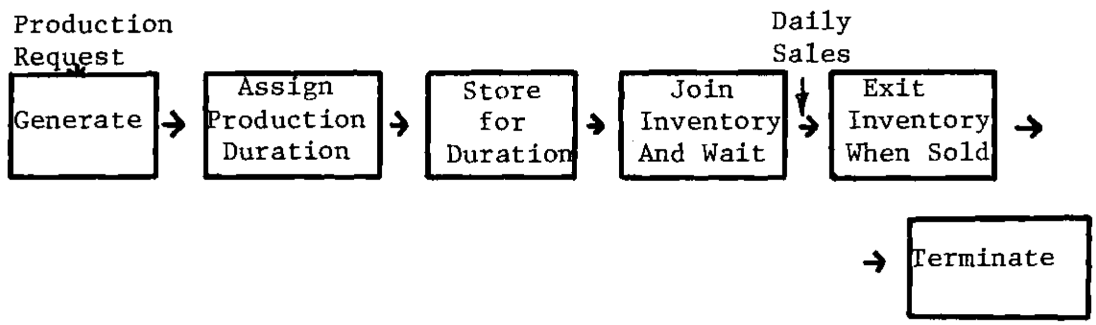


Figure 47. Discrete Formulation

An important modeling concern is the resolution of time intervals: into how many slots ought the production time be disaggregated? In this system, a one day fundamental time unit is warranted, even though the production department may complete items throughout a day. Recall that the control mechanism samples sales and inventory only at the end of each day. Since the control structure is indifferent to exact timing of additions or deletions to inventory, and since model performance is measured via daily variance, a smaller FTU is not needed. The two item-by-item models are named model I (with constant sales) and model IS (with noisy sales). Appendix 3 contains listings for the item-by-item models.

5.2.2 Discrete Model with Constant Sales

Model "I" is initialized with twenty units in production, twenty in inventory, and run for 150 days. Using ten different seeds for the random number generators, ten replicates are produced. The performance measure, daily variance, range from 18.25 to 26.01; average variance is 21.58.

All simulated output series (see Figure 48 for sample series) display two qualitative patterns. First, starting from its initial value of 20, inventory experiences overshoot -- the goal of 40 units is exceeded. After about 40 days, the inventory settles and randomly hovers about the equilibrium value. Second, quasi-periodic patterns are within the random hovering. This oscillatory behavior has a period of about 15-16 days and extends throughout the output series. The random number generator (and seed values) was carefully checked for periodic behavior; none was apparent. The oscillation is latent in the system control structure, but a GPSS-like diagram, as Figure 47, hides that structure.

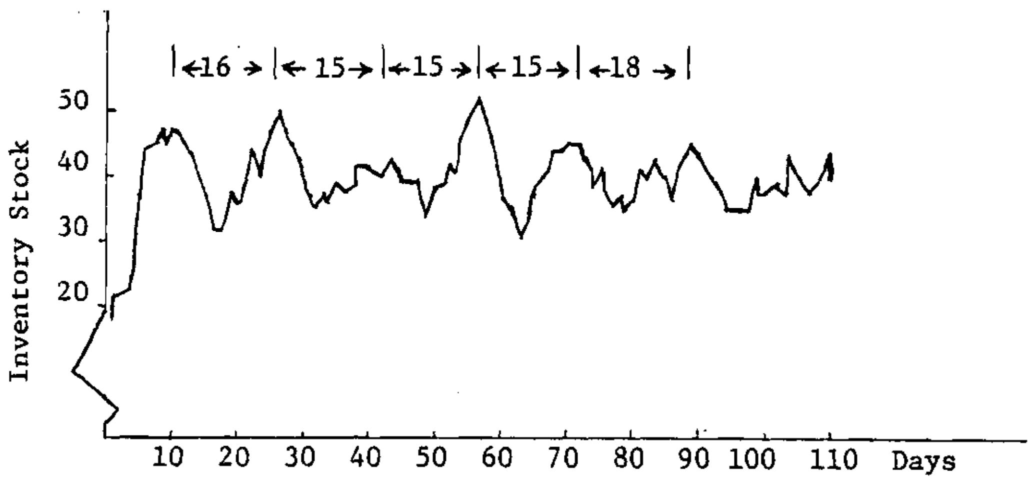


Figure 48. Sample Output from Model I

5.2.3. Discrete Model with Stochastic Sales

Model I is modified to become model IS. The value of daily sales in model IS is a clipped draw from a normal distribution with mean 4 and variance 3.5 -- actual sales are constrained to be non-negative, integral, and less than the inventory stock. Integrality is imposed on the draw by the following method: The draw is truncated and then given a [(draw - truncated value) * 100]% chance of being incremented by one.

As expected, model IS generates greater variance than does model I. Average variance (ten replicates) is 41.58, nearly twice the estimate for model I. Model IS output trajectories show oscillatory behavior with period of 11 to 20 days. The behavior is less regular, but qualitatively similar to model I output.

5.3 Solution Using Continuous Formulations

Armed with the discrete models we can test the adequacy of naive continuous simulation techniques. But first we need to develop the appropriate continuous models.

5.3.1 Continuous Model Design

The first stage in formulating a continuous model is creating an influence diagram indicating the cause and effect structure of the system, shown in Figure 49.

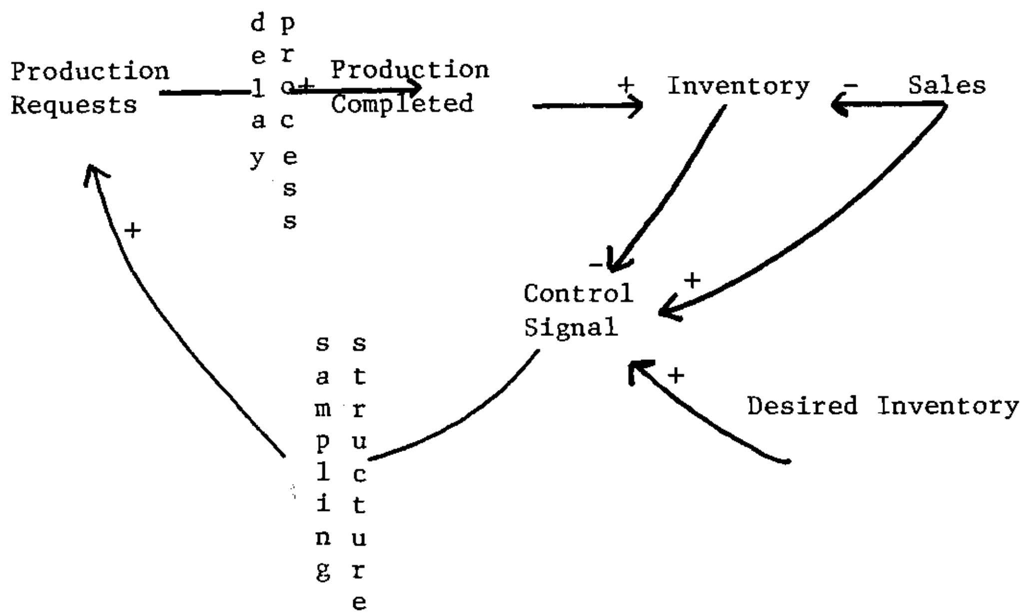


Figure 49. Influence Diagram of System

Arrows on Figure 49 point from a causing variable to an affected variable. The sign of an arrow indicates the direction of effect. For example, the negative arrow from sales to inventory indicates that sales causes inventory to decrease. Similarly, the positive arrow from production-completed to inventory indicates that the number of units leaving production increases the number of units in inventory.

Figure 49 shows the use of an exponential delay structure to mimic the production lag, i.e., the "delay process" between production request and production completion. The reorder policy is also included in Figure 49. The values of inventory, sales and desired inventory combine via equation (5-1) to form a control signal. To mimic the daily (rather than continuous) evaluation of reorders, the control signal passes through a sampling structure before activating production requests.

A state space representation of the influence diagram requires two integral equations. Inventory is the integral of inflow minus outflow, i.e., production-completed minus sales.

$$I(t) = I_o + \int_0^t [\text{Prodcomp}(s) - \text{Sales}(s)]\, ds \tag{5-2}$$

The production process is modeled as a first order exponential delay with mean throughput time of 4 days. Equations for production-completed and in-process accumulation are

$$\text{Inprocess}(t) = \text{Inprocess}_o + \int_0^t [\text{Requests}(s) - \text{Prodcomp}(s)] ds \tag{5-3}$$

$$\text{Prodcomp}(t) = \text{Inprocess}(t-1)/4.0 \tag{5-4}$$

Figure 50 presents the overall continuous model in Feedback Dynamics notation.

The computer implementation for this model uses Euler integration. A solution step size "DT" for the Euler scheme is required. This step size is not directly related to the fundamental time unit -- DT is an artifact of numerical integration. Suggestions for the size of DT may be found in the literature. Forrester recommends DT on the order of half of the smallest delay time constant (here 1/2 of 4 days)[1]. By selecting

---

[1] Forrester [1961], page 124.

DT = 1 the sampling in the control loop is handled implicitly.

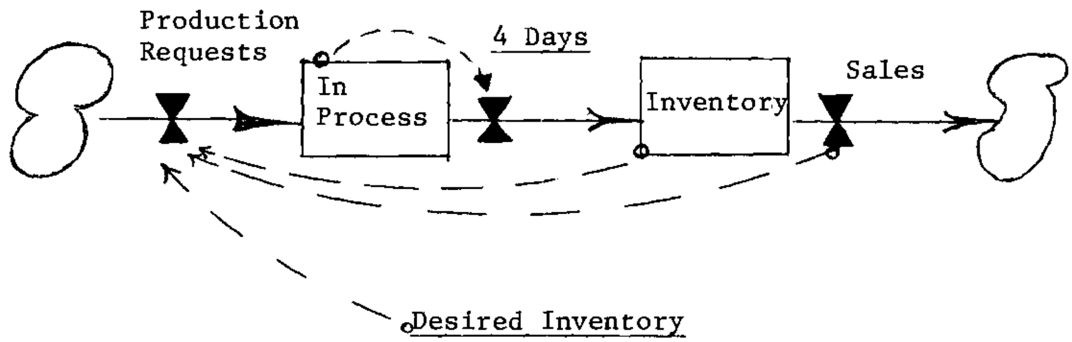


Figure 50. Continuous Formulation

5.3.2 Continuous Model with Constant Sales

The model (see Appendix 4 for listings of all continuous models) with constant sales is initialized with twenty units in production, twenty in inventory, and run for 150 days. The performance of this model (Model "C" - with constant sales) yield an inventory variance of 13.5. The inventory trajectory experiences initial overshoot, with inventory reaching 50.8 units, followed by dampened oscillations about the 40 unit desired value. The period of oscillation is 18 days, and quickly dampens: the series of peaks is 50.8, 43.1, 41.0, 40.3, and 40.1. After ninety days, the model is in steady state (see Figure 51).

For linear dynamic systems, such oscillatory behavior can be mathematically predicted[1]. The magnitude and period of oscillation depend on the gain (the "1/2" in equation (5-1)), delay width and delay order. In the face of a disturbance or initial disequilibrium, the inventory level oscillates while attempting to regain equilibrium -- a characteristic

---

[1] Brown and Nilson [1962], page 142.

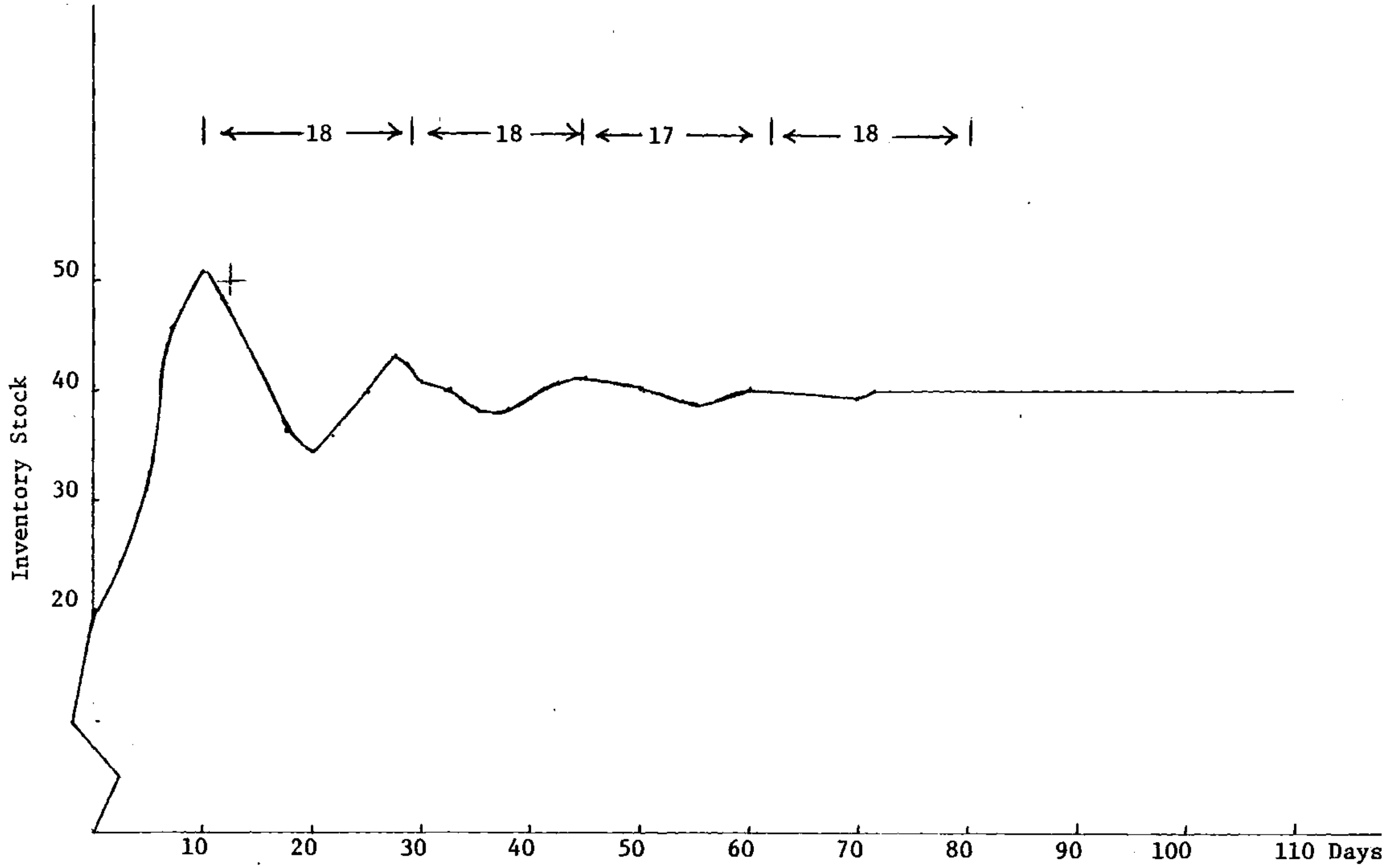


Figure 51. Output from Model C

of second order linear differential systems.

5.3.3 Continuous Model with Stochastic Sales

Model C is appropriately modified to become model CS, the random sales variant of the inventory system. The value of daily sales is a clipped draw from a normal distribution of mean 4 and variance 3.5. The draw is clipped to be nonnegative and less than the current inventory level. Model CS is given the same starting conditions as model C, and exercised with 10 different random number seeds. The ten replicates yield an average variance of 38.8, as compared with 13.5 for model C.

The inventory trajectories of model CS resemble those of model C, (see Figure 52 for a sample series) except the oscillations are not deterministic. The random nature of sales does not let the feedback control close exactly to the desired inventory value. Oscillation is present in the output series throughout the entire run of the simulation whereas model C reached steady state by time 90. The oscillations lack perfect regularity, but are clearly present (see Figure 52).

Not all simulated series contain pronounced oscillation. An analysis of those series which did oscillate indicated that the control system of the inventory would react to a rare (i.e., a large or small) random sales value by initiating and propagating an oscillatory reaction. If the random values used in a particular run are all close to the average value, then the system had no reason to produce oscillations.

5.4 Difference in Performance

The upper half of Table 7 summarizes the results of the four models just considered. In both sales rate variants, the continuous

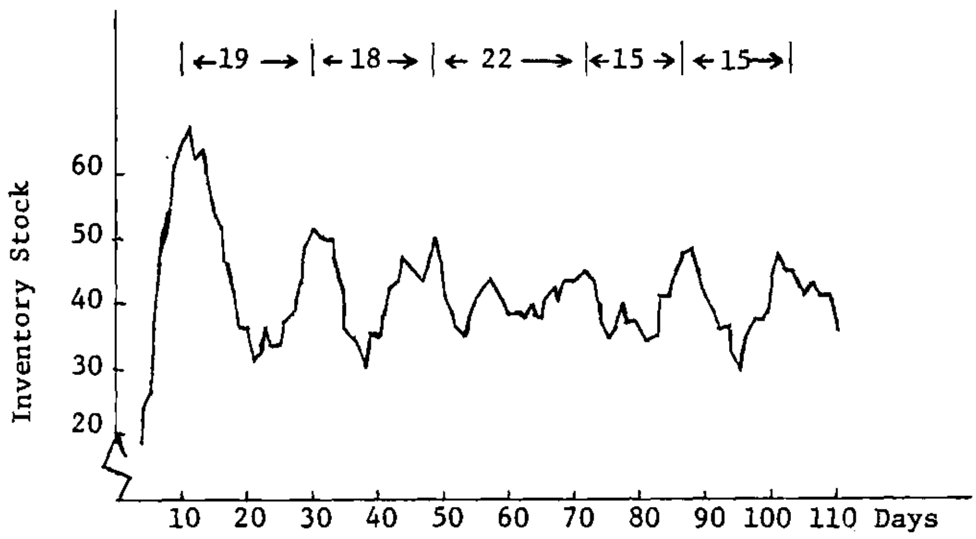


Figure 52. Sample Output from Model CS

formulations produce less output variance than the corresponding discrete model.

The discrepancy between formulations is most evident in the constant sales system. The discrete model produces a variance of 21.6 whereas the continuous model produces a variance of 13.5. A comparison of output trajectories depicted in Figures 48 and 51 illustrates that the naive continuous model does not accurately represent the referent system. This is evident from the randomness in Figure 48 and the deterministic dampened oscillations of Figure 51.

Since a change in delay formulation (from discrete to continuous) does impact model performance, we wish to determine what can be done to lessen the impact of formulation interchange. The next section investigates two potential corrective techniques.

5.5 Corrective Techniques

There are numerous adjustments which could be imposed upon the naive continuous models to improve their performance. Below are presented two schemes for bringing the variance of continuous models closer to that of their real world discrete referents. Both schemes rely on the incorporation of a noise source within the model. The issues at hand are where to embed the noise, what variance should the noise have, and what time structure should the noise have.

The results of Chapters 3 and 4 may be exploited to help answer these questions. The bottom half of Table 7 lays out two corrective schemes: artificial noise incorporated on sales, and artificial noise incorporated in the delay output. They are considered in turn.

5.5.1 Incorporation of Artificial Noise onto Sales

The real world system here is captured in Model I -- deterministic sales rate, discrete exponential production delay. The naive continuous approximation (model C) gives average variance of 13.5, not 21.6. What's missing from model C is Var_{dp} -- the variance due to the probabilistic selection of delay lags by discrete entities. We know what this variance should be. Chapter 3 shows the output variance of a discrete delay to be approximately γN. In this case, for an exponential delay of mean 4, γ is .87; N for the system is 4 units per day. The expected variance in output from the item-by-item delay in model I is (.87)(4) or 3.5. Model C* using the first corrective scheme adds a 3.5 unit^2 variance white

Table 7. Summary of Inventory System Formulations

	Referent: Inventory System with Deterministic Sales				Referent: Inventory System with Stochastic			
Basic Models:								
Name	Production Delay	Sales	Variance	Name	Production Delay	Sales	Variance	
I	random (causal)	detrm.	21.6	IS	random (causal)	random (causal)	41.6	Real World
C	state space	detrm.	13.5	CS	state space	random (causal)	38.8	Naive Continuous Approximations
Corrected Models								
C*	state space	random (artif.)	38.8		does not make sense			
C+	random (artif.)	detrm.	25.5	CS+	random (artif.)	random (causal)	43.2	Improved Continuous Approximation

noise disturbance to the system's sales rate.

This correction is artificial: sales is supposed to be deterministic. Some continuous system modelers believe that adding noise at a nearby point in a complex control system serves as well as adding stochastic elements anywhere else. Model C* allows a check of this assertion. Incidentally, for this problem's parameters (4 unit/day sales, $\sigma^2 = 3.5$ for varying sales case, 4 day exponential delay), it turns out that corrective model C* (which tries to patch the continuous formulation of the deterministic sales variant) is the same as model CS (the naive continuous formulation for the stochastic sales variant). This is a planned occurrence which reduces the number of simulation runs required, but in general, model C* is not identical to model CS. In any case, Table 10 indicates that model C* does not perform similarly to model I, variance of 38.8 versus 21.6.

Embedding variability anywhere in a continuous model is not enough. In this inventory example, the deviation in performance between model C* and model I is caused by not properly considering autocorrelation. Chapter 3 illustrated the autocorrelation in the outflow of discrete delays. In particular, Table 4 defines the expected autocorellation in the output of an exponential delay of mean 4. (With FTU equal to one day, this delay has width of 4.) The autocorrelations are -.11, -.09, -.07 and -.05 for the first four lags.

Through the use of time series techniques[1], a noise source with autocorrelated behavior could be created. This source can then be

---

[1] Box and Jenkins [1970]

incorporated in the calculation of sales in a fashion similar to model C*. But a simpler method yielding good results is available.

5.5.2 Incorporation of Artificial Noise onto Delay Output

This corrective technique uses a white noise sequence added to delay output. White noise may be added to the output of any continuous delay so that the adjusted process approximates much more closely the delay's real world referent in both variance and autocorrelation structure.

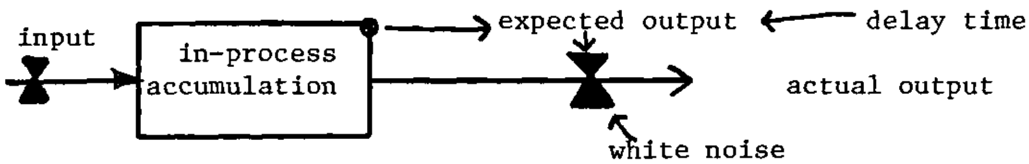


Figure 53. Corrective Technique C+

Actual delay outflow is the expected outflow plus a draw from a zero mean white noise generator. The conservative nature of the delay process forces negative autocorrelation in the output series. A high output now, leaves less in the delay to exit later with a good chance of lower than expected output, and vice-versa. Further, the delay rapdily "forgets" past outputs. So we have the correct qualitative pattern derived in Chapter 3 -- negative autocorrelation of smaller and smaller magnitude imparted to larger lags.

For example, when the delay content is 20 units, the expected outflow is (by equation (4-2)) 5 units. Should the noise generator yield +3, eight units exit, leaving only twelve (rather than the expected 15) units in the delay. Assuming no input to the delay, only 3 units are expected to leave during the next time period, rather than 3.75 if the random draw had been zero. The short lag negative autocorrelation is clear.

5.5.2.1 System with Constant Sales - Model C+. We incorporate this noise source into model C to let it become a corrected model C+. With the same starting conditions as in the other models, model C+ produces average variance of 25.5 over a sequence of 10 replicates. This value is certainly closer to the 21.6 variance for model I compared with model C*'s value of 38.8.

5.5.2.2 System with Random Sales - Model CS+. Here the real world referent is model IS -- a discrete delay with stochastic sales rate. Model CS+ attempts to improve on CS, the naive continuous formulation for IS. CS+ is a continuous model with stochastic sales (the correct variance) and a noise source within the delay structure. The noise is incorporated in the same fashion as indicated in Figure 53. The average variance from ten replications of CS+ is 43.2, closer to the real referents 41.6 average variance than is the naive continuous formulation's performance.

Recall that model CS, without any "corrective" technique produces a variance estimate of 38.8, a value also close to the performance of model IS. Model CS is more straightforward, and executes faster than model CS+, but model CS's performance is 4.4 units^2 from the item-by-item referent, whereas model CS+'s index is 1.6 units^2 from the referent. However, since all three variance values 38.8 (model CS), 43.2 (model CS+), and 41.6 (model IS) are estimates of the expected value of a random variable, it is difficult to judge model superiority. Here the naive continuous formulation may be acceptable.

5.6 Conclusions

This inventory-production example shows that continuous models with state space delays may be modified to closely emulate discrete real world referents. What is required is that the analyst recognize the variance and autocorrelation properties of discrete delays. This example also provides support for part of the claims of continuous modelers. Under some conditions, real world systems with discrete stochastic delays _and_ stochastic flows may be approximated by continuous models with approximate state space delays and correctly modeled stochastic flows.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

The objective of this thesis is to quantitatively compare the behavior of state space surrogates used in continuous simulation with the behavior of discrete real world delay processes. The continuous and discrete delay processes are contrasted on three levels: structure, component behavior, and whole model behavior. This chapter summarizes the finding in three sections corresponding to the three levels of contrast, and ends with a section of recommendations for future study.

6.1 Comparison of Structure

This section contrasts the structure of the delaying mechanism in discrete and continuous delays.

6.1.1 Delay Mechanism - Discrete

A discrete delay process as found in the real world is probabilistic, conservative, and operates on discrete entities. Figure 54 depicts the structure of a discrete delay.

Each discrete entity of an input flow individually draws an "in-process" lag from the delay's lag distribution. The entity then resides in the delay's internal accumulation from the chosen "in-process" lag. The discrete delays considered in the thesis are further restricted to have entity entry and entity exit occur during equally spaced time intervals. The spacing is called the fundamental time unit (FTU). Given a fundamental time unit, the probability structure for an item by item

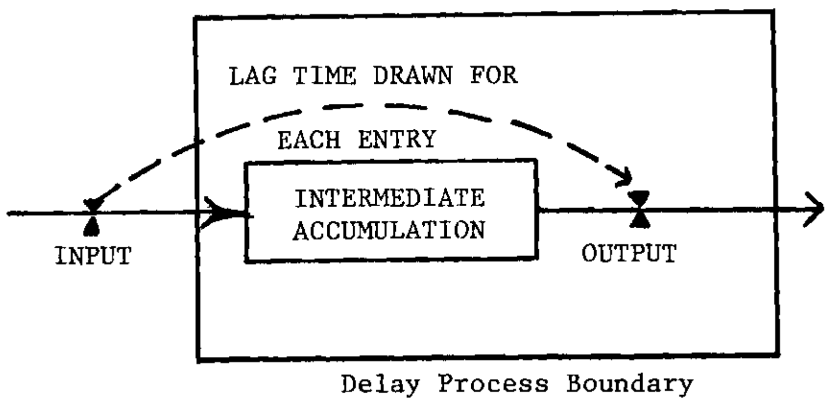


Figure 54. Delay Process

delay is a discrete distribution. The number of choices in the distribution is termed delay "length".

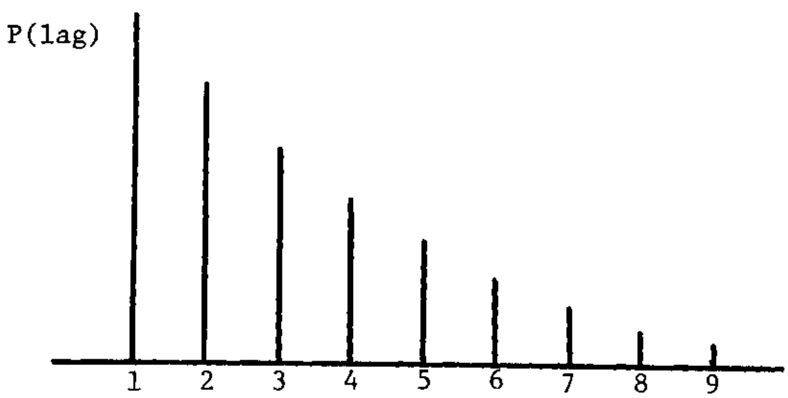


Figure 55. Discrete Exponential Distribution

Figure 55 illustrates a discrete representation of an exponential distribution with length 9.

When used in conjunction with infinite distributions, such as exponential, "length" is arbitrary, depending on the accuracy desired in the discrete representation. A more descriptive measure is delay "width", a word used here to represent distribution dispersion -- the range of a uniform distribution, the standard deviation of a normal distribution, and the mean of the exponential and Erlang3 distributions.

6.1.2 Delay Mechanism - Continuous

A continuous delay surrogate is used to approximate an item by item delay. Most generally we can view these surrogates as impulse response functions which operate on an input stream to yield an output stream. The delay surrogates here considered are a subset using a small number of linear differential equations of the following form:

$$\dot{X} = AX + Bu \tag{6-1}$$

$$y = CX$$

where X is a vector of the contents of the accumulations of the delay, A is an n by n matrix, B and C are vectors, and y and u are the scalar output and input respectively. More particularly for an nth order exponential delay, A, B, and C, must follow the pattern below:

$$A = \begin{bmatrix} \frac{n}{D} & 0 & 0 & \cdots & 0 \\ \frac{n}{D} & -\frac{n}{D} & 0 & \cdots & 0 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & 0 & 0 & \cdots & \frac{n}{D} \; -\frac{n}{D} \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \qquad C = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \frac{n}{D} \end{bmatrix} \tag{6-2}$$

A surrogate of this form has an Erlang-shaped impulse response function as shown in equation (6-3).

$$IRF(T) = \frac{T^{n-1} e^{-Tn/D} n^n}{(n-1)!\ D^n} \tag{6-3}$$

Here, n is the order of the delay, and D is the mean throughput time for the entire delay surrogate.

6.1.3 Delay Mechanisms - Comparison

The continuous delay is designed to be conservative (as is the discrete delay), but it is neither probabilistic nor discrete. Even though the impulse response functions are shaped like Erlang probability functions, the operation of the continuous delay is probabilisticly degenerate. Each input pulse is deterministicly broken down into an output train identical to the delay's impulse response function. This often forces fractional entities to exit the delay, thus destroying the integral nature of the discrete entities.

Another structural difference is that real world delays can have arbitrary probability functions whereas the continuous delay surrogates here considered are restricted to Erlang-shaped impulse response functions. Hence the expected behavior of state space surrogates cannot always be equated to the expected behavior of the referent real world delay.

There is clearly a difference in the structure of discrete and continuous delays. But do the structural differences produce important differences in behavior? And are behavior differences important for overall model performance?

6.2 Comparison of Behavior

To ascertain the importance of the structural differences between discrete and continuous delay formulations for component behavior, the thesis examines the output from each delay formulation under conditions of identical input. Three forms of input are considered: constant, sinusoid and white noise.

6.2.1 Output Behavior - Discrete Delay

The stochastic nature of lag selection in a discrete delay creates stochastic output. The thesis derives statistical measures of the stochastic behavior of the output series.

Constant Input. Given constant input, a discrete delay produces output whose mean, variance and autocorrelation are found to be

$$\mathrm{Mean}[O(t)] = N \tag{6-4}$$

$$\mathrm{Var}[O(t)] = N \sum_{i=1}^{a} [P(i)\ (1-P(i))] \tag{6-5}$$

$$\mathrm{Autocorrelation}[\mathrm{lag}\ i] = \frac{\sum_{j=1}^{a-i} [P(j)P(j+i)]}{1 - \sum_{j=1}^{a} P(j)^2} \quad \text{for } i < a \tag{6-6}$$

$$= 0 \qquad \text{for } i \geq a$$

Here N is the input magnitude (i.e., average), P(.) the delay lag distribution and "a" the length of the delay lag distribution.

The influence of the input magnitude or delay distribution on these measures may be derived. Both the mean and variance of the output vary linearly with the input magnitude N. Autocorrelation is not a function of input magnitude.

The delay lag distribution affects variance and autocorrelation, but not mean. It is useful to define a descriptor of the probability distribution -- the summation γ:

$$\gamma \triangleq \sum_{i=1}^{a}[P(i)\ (1-P(i))] \tag{6-7}$$

From equation (6-5), γ is seen to be the ratio of output variance to input magnitude. Section 3.3.3 shows that γ is bounded by 0 and 1. Wide distributions (i.e., with large standard deviations) create γ values close to unity. Conversely, narrow distributions create summation values closer to zero.

The distribution's shape also affects autocorrelation values. Section 3.3.4 illustrates that wide distributions produce autocorrelations of negative sign and small absolute value. Narrow distributions also produce negative autocorrelation values, but the values are of greater absolute value (where non-zero).

Sinusoidal Input. Given sinusoidal input, a discrete delay produces output O(t) which follow equations (6-8) and (6-9).

$$E[O(t)] = \sum_{i=1}^{a}[I(t-i)P(i)] \tag{6-8}$$

$$\mathrm{Var}[O(t)] = \sum_{i=1}^{a} [I(t-i)P(i)(1 \quad P(i))] \tag{6-9}$$

Here I(t) is the input at time t, P(.) the delay lag distribution, and "a" the delay distribution length.

Equations (6-8) and (6-9) are valid for any form of input, and are utilized in Section 4.2.2 to analyze discrete delay behavior given sinusoid input. It is shown that the expected output is a sinusoid with frequency equal to the input signal, and amplitude less than the input signal. The variance of the realized output depends upon the delay distribution shape and width, and also upon the height of the carrier (the average input value). Greater distribution width, and larger average input produces greater output variance which can mask the periodicity of the expected output signal (see section 4.3.1).

<u>White Noise Input</u>. With white noise input discrete delays produce output whose variance is given by equation (6-10).

$$\mathrm{Var} = \gamma N + \sigma^2 (1-\gamma) \tag{6-10}$$

Here σ^2 is the input variance, N the average input, and γ is a function of the delay lag distribution:

$$\gamma = 1 - \sum_{i=1}^{a} P^2(i) \tag{6-11}$$

A discrete delay "funnels" the output variance (see Figure 46) to be close to the value of N. High input variance (greater than N) yields output variance less than the input variance but greater than N; and low input variance (less than N) yields output variance greater than the input variance but less than N.

6.2.2 Output Behavior - Continuous Delays

The analytic nature of continuous delays produce deterministic and analytic output. The output is defined by equation (6-12).

$$O(t) = \int_0^\infty I(t-T) \frac{T^{n-1} e^{-Tn/D} n^n}{(n-1)!\ D^n} \, dT \tag{6-12}$$

where n is the order of the delay approximation, D the mean throughput time, and I(t) the input at time t. The output for the three forms of input are explained below.

Constant Input. For constant input, a continuous delay produces output which (after an initial warm-up period) is equal to the input. This is true for all orders of state space delay surrogates.

Sinusoidal Input. For sinusoidal input, a continuous delay produces output which is sinusoidal, of the same frequency as the input signal, but with reduced amplitude and phase shift. The amplitude attenuation and phase shift properties are presented in Figures 31 and 32 of Chapter 4.

White Noise Input. For white noise input, a continuous delay produces an output signal which has structure in time (i.e., the output signal is non-white) and which has variance less than the input signal.

The high frequency components of the input signal are filtered from the output signal according to the attenuation property of the particular surrogate.

6.2.3 Differences in Behavior

Continuous delays act like smoothing filters. Delay output is always smoother (that is, with less variance) than the input signal. This is not the case in discrete delays. The stochastic nature of lag selection produces a noisy output series even when the input series is constant.

The continuous delay behavior is very similar to the "expected" behavior of the discrete delay. For instance, the attenuation property of an Erlang3 discrete delay (as calculated from the expected output series) is equivalent to the attenuation of a continuous Erlang3 delay (see section 4.2.2 for more discussion of this topic).

An important behavior difference resulting from the common practice of using third order (Erlang) continuous delays as surrogates for real world normal delays is derived in section 4.3.2. The expected attenuation of the real world normal delay can be substantially different from the continuous surrogate's attenuation.

The impact of component behavior differences on overall model performance is hard to gauge. The feedback loop structure of the model may mask or amplify the stochastic component of discrete delay behavior. The final area of analysis considers this issue.

6.3 Impact of Component Error on Model Behavior

The significance of continuous surrogate error on model behavior is case dependent. The error is a function of the specific model in where the delay approximation is embedded. The inventory-production example of Chapter 5 illustrates that under some conditions real systems with discrete delays and stochastic flows may be approximated with small behavior difference by continuous models with state space delays and correctly modeled stochastic flows.

In systems with deterministic flows, the continuous delay formulation can cause major differences in overall model behavior. The performance of the continuous model can be improved, however, with the incorporation of a white noise source (see section 5.5 for details of noise incorporation) into the calculation of delay outflow.

6.4 Recommendations for Future Study

The analysis presented in this thesis addresses only state space delay surrogates, and only steady state input. The following paragraphs recommend areas of future research.

Delay surrogates, other than the nth order exponential components considered in the thesis can be used in continuous models. These components allow impulse response functions with the form of Erlang probability distributions. Impulse response functions of more general shapes[1] may be easily built. However, they require more state variables and elements of memory than do the exponential delays. Other surrogates

---

[1] The autoregressive, moving-average filters of time series analysis described in Box and Jenkins [1970] are an example.

with state-dependent lag distributions may better represent the real world delay. Future study can assess the utility of these more complex structures and quantify their behavior.

Different performance measures can be addressed for the more complex surrogates. For example, temporal variance i.e., the variance of throughput times, is an important descriptor for the state-dependent processing surrogates.

The treatment of input has been limited to steady state flows. Two categories of input are not considered: transient and non-stationary. Both transient and non-stationary input flows are important due to their prevalance in modeling situations, and thus deserve investigation.

A more extensive analysis of surrogate impact on overall model adequacy can be undertaken. The first phase of the analysis would survey the performance measures used in modeling exercises. The survey could yield an indication of the percentage of models whose performance measure was: an end-of-run value of a flow or accumulation, the average value of a flow or accumulation calculated over the entire simulation, the variance of a flow or accumulation about a certain value, etc. A related survey could categorize the form of input which enters a delay during model execution. Then one should study surrogate performance focusing on the more often encountered performance measures and the more often encountered input flows.

The design of corrective techniques deserves more research. The survey of total model performance measures may indicate that measures other than variance and attenuation (which are covered in this thesis) need attention.

It is difficult to prioritize these research activities. The priority of analysis should be related to the impact of analysis results on the simulation field. With this in mind, the most needed research is the survey of modeling usage to determine the use of complex delaying processes, the forms of performance measures, and the forms of input entering delaying processes. This survey would indicate the areas of greatest research need.

A P P E N D I C E S

APPENDIX I

PROGRAMS TO CREATE VARIANCE AND AUTOCORRELATION

This appendix contains the Fortran programs which calculate the variance and autocorrelation values which appear as Tables 1, 3, 4, 5 and 6. There is a unique program for each of the four distributions: uniform, normal, exponential and Erlang3. The logic of the four programs is identical save for the calculation of the P(.) array defining the delay lag distribution. A single program will calculate either autocorrelation or variance depending on the initial user input value.

Program Logic:

1. Read input values which select either variance or autocorrelation computation.
2. Define delay width.
3. Using delay width, calculate P(.) array.
4. Using P(.) and equation 3-15, determine variance.
5. Using P(.) and equation 3-38, determine autocorrelation.
6. Output values.
7. Go to #2 to perform calculations for a large width.

```
D-H*TK2.UNI
   C   THIS PROGRAM CALCULATES GAMMA AND
   C   AUTOCORRELATION FOR CONSTANT INPUT INTO UNIFORM
   C   DELAY
   C
   C   PR(I) IS THE PROBABILITY OF CHOOSING A LAG OF I FTU'S
   C   AUTO(K) IS THE AUTOCORRELATION FOR LAG K
   C   LENGTH IS THE RANGE OF THE UNIFORM DELAY
   C   IOP = 1 WILL CAUSE PROGRAM TO PRODUCE AUTOCORRELATIONS
   C   IOP = 2 WILL CAUSE THE PROGRAM TO PRODUCE GAMMA VALUES
          DIMENSION PR(251),AUTOC(251)
          WRITE(6,1)
   1      FORMAT('  UNIFORM GAMMA / AUTOCORRELATION')
          PRINT  299
   299    FORMAT(' ENTER 0 FOR GAMMA, 1 FOR AUTOCORRELATION')
          READ(5,298) IOP
   298    FORMAT(I1)
          IF(IOP.EQ.1)PRINT 300,(I,I=1,10)
   300    FORMAT(1H1,'         EXPECTED AUTOCORRELATIONS FOR UNIFORM DELAYS'
         * ,//,23X'LAG:',/,' LENGTH',10I5)
   C
   C PERFORM CALCULATION FOR A SERIES OF DELAY LENGTHS
          DO 200 LENGTH=2,20
   C   WE NEED TO CALCULATE THE 'PR' PROBABILITIES
   C   SP WILL BE THE SUM OF ALL PR(IX)'S AND USED TO CHECK
   C   PR(IX) CALCULATIONS.
          SP=0.
          GAMMA=0.
          DO 100 IX=1,LENGTH
          P=1./FLOAT(LENGTH)
          SP=SP+P
          PR(IX)=P
          GAMMA=GAMMA+P*(1.-P)
   100     CONTINUE
          IF(SP.LT.0.99) PRINT 117,SP
   117    FORMAT(' ERROR - SUM OF PROBABILITIES IS ',F6.3)
   C
   C   WE CAN NOW GENERATE COVARIANCES
   C   INDEX I POINTS TO THE I-TH LAG COVARIANCE
   C   INDEX J IS USED TO INDICATE COMPONENTS SHARED IN THE
   C    VARIABLES SEPARATED BY I LAGS
          IM1=LENGTH-1
          DO 45 I=1,IM1
          COV=0.
          MI=LENGTH-I
          DO 40 J=1,MI
   40     COV=COV+PR(J)*PR(J+I)
   45     AUTOC(I)=-COV/GAMMA
   C
   C   ALL DONE   TIME FOR PRINTING
          IF(IOP.EQ.0)PRINT 110,LENGTH,GAMMA
   110    FORMAT(' LENGTH=',I5,'    VAR/N=',F6.2)
   C  FOLLOWING LOGIC PREVENTS PRINTING OF ZEROS.
          IP=10
          DO 301 I=1,10
          IF(AUTOC(I).LT.-0.005) GOTO 301
          IP=I
```

```
      GO TO 302
301   CONTINUE
302   IF(IOP.EQ.1)PRINT 305,LENGTH,(AUTOC(I),I=1,IP)
305   FORMAT(3X,I2,3X,10F5.2)
C
C  RETURN TO DO CALCULATION FOR THE NEXT LENGTH DELAY
200   CONTINUE
      STOP
      END
```

?.NOR

```
@D-H*TK2.NOR
C    THIS PROGRAM CALCULATES GAMMA AND
C   AUTOCORRELATION FOR CONSTANT INPUT INTO NORMAL
C   DELAY
C   PR(I) IS THE PROBABILITY OF SELECTING A LAG OF I FTU'S
C   AUTOC(K) IS THE AUTOCORRELATION FOR SEPARATION K
C   SIGMA IS THE STANDARD DEVIATION FOR THE PROCESS
C   LEN IS THE LENGTH  (OR NUMBER OF SLOTS) OF THE DELAY
C       DISTRIBUTION       ===   6 TIMES THE SIGMA
C   IOP=1 CAUSES AUTOCORRELATION TO BE PRINTED
C   IOP=0 CAUSES GAMMA TO BE PRINTED
C
C   GATHER INPUT DATA......
        DIMENSION PR(251),AUTOC(251)
        INTEGER SIGMA
        WRITE(6,1)
1       FORMAT('  NORMAL GAMMA / AUTOCORRELATION')
        PRINT  299
299     FORMAT(' ENTER 0 FOR GAMMA, 1 FOR AUTOCORRELATION')
        READ(5,298) IOP
298     FORMAT(I1)
        IF(IOP.EQ.1)PRINT 300,(I,I=1,10)
300     FORMAT(1H1,'      EXPECTED AUTOCORRELATIONS FOR NORMAL DELAYS'
       * ,//,23X'LAG:',/,'    SIGMA',10I5)
C
C   DO CALCULATIONS FOR DIFFERENT VALUES OF SIGMA
        DO 200 SIGMA=1,20
C   WE NEED TO FILL UP THE PR VECTOR WITH PROBABILITIES
C   THIS WILL BE DONE THROUGH USE OF THE CUMULATIVE DENSITY
C   FUNCTION FOR THE NORMAL DISTRIBUTION
C   FN IS THE VALUE OF THE CUMULATIVE AT THE POINT OF INTEREST
C   F0 IS THE PREVIOUSLY ACCESSED VALUE OF THE CDF
C   THUS THE PROBABILITY IS FN-F0
C   SP WILL WILL THE SUM OF THE PR VECTOR.. USED TO CHECK
C       THE ACCURACY OF PR GENERATION
        F0=0
        LEN=6*SIGMA
        GAMMA=0.
        IMEAN=3*SIGMA
C   IMEAN IS THE MEAN OF THE DISTRIBUTION
        SP=0.
        DO 100 IX=1,LEN
        FN=ZVAL(FLOAT(IX-IMEAN)/FLOAT(SIGMA))
        P=FN-F0
        SP=SP+P
        F0=FN
        PR(IX)=P
        GAMMA=GAMMA+P*(1.-P)
100      CONTINUE
        IF(SP.LT.0.99) PRINT 117,SP
117     FORMAT('  PROBABLE ERROR- SUM OF PROBABILITIES IS',F6.3)
C
C   WE CAN NOW GENERATE AUTOCOVARIANCES
C   INDEX I INDICATES THE SEPARATION LAG
C   INDEX J POINTS TO COMPONENTS SHARED BY PULSES ON LAG I
        LEN1=LEN-1
        DO 45 I=1,LEN1
```

```
      COV=0.
      MI=LEN-I
      DO 40 J=1,MI
40    COV=COV+PR(J)*PR(J+I)
45    AUTOC(I)=-COV/GAMMA
      IF(IOP.EQ.0)PRINT 110,SIGMA,GAMMA
110   FORMAT(' SIGMA=',I5,'  VAR/N=',F6.2)
C  FOLLOWING LOGIC  STOPS THE PRINTING OF ZEROS
      IP=10
      DO 301 I=1,10
      IF(AUTOC(I).LT.-0.005) GOTO 301
      IP=I
      GO TO 302
301   CONTINUE
302   IF(IOP.EQ.1)PRINT 305,SIGMA,(AUTOC(I),I=1,IP)
305   FORMAT(3X,I2,3X,10F5.2)
C
C  GO TO NEXT VALUE OF SIGMA
200   CONTINUE
      STOP
C
      FUNCTION ZVAL(X)
      INTEGER HIGH
C  DETERMINE CUMULATIVE PROBABILITY OF N(0,1) AT THE
C  POINT X
C  VECTOR VAL HOLDS CUMULATIVE VALUES FOR THE NORMAL
C  IN TERMS OF .1 SIGMAS....     VAL(1) IS THE CUMULATIVE
C  AT -.0 SIGMA       VAL(I) IS THE CUMULATIVE AT
C  THE POINT -(I-1)/10 SIGMA
      DIMENSION VAL(31)
      DATA VAL/.5000,.4602,.4207,.3821,.3446,
     *  .3085,.2743,.2420,.2119,.1841,
     *  .1587,.1357,.1151,.0968,.0808,
     *  .0668,.0548,.0446,.0359,.0287,
     *  .0228,.0179,.0139,.0107,.0082,
     *  .00621,.00466,.00347,.00256,.00187,.000968/
C  ASSUME 'X' WAS NEGATIVE
      Y=ABS(X)*10.+1
C  NOTE THAT THE *10 CHANGES METRIC TO .1 SIGMAS AND THE
C  +1 SHIFTS ORIGIN SINCE VAL(1) IS FOR A .0 SIGMA VALUE
C  THE CORRECT VALUE FOR Y LIES BETWEEN LOW AND HIGH
      LOW=Y+1
      HIGH=Y
      ZVAL=0.
      IF(LOW.LE.30) ZVAL=VAL(LOW)+(LOW-Y)*(VAL(HIGH)-VAL(LOW))
C  IF 'X' WAS NOT NEGATIVE, CORRECT ZVAL VALUE
      IF(X.GT.0.) ZVAL=1.-ZVAL
      RETURN
      END
```

.EPL

```
U-H*TK2.EXP
C    THIS PROGRAM CALCULATES GAMMA AND
C   AUTOCORRELATION FOR CONSTANT INPUT INTO EXPONENTIAL
C   DELAY
C   PR(I) IS THE PROBABILITY OF SELECTING A LAG OF I FTU'S
C   AUTOC(K) IS THE AUTOCORRELATION FOR SEPARATION K
C   MEAN IS THE EXPECTED THROUGHPUT TIME
C   LEN IS THE LENGTH  (OR NUMBER OF SLOTS) OF THE DELAY
C      DISTRIBUTION      ===    6 TIMES THE MEAN
C   IOP=1 CAUSES AUTOCORRELATION TO BE PRINTED
C   IOP=0 CAUSES GAMMA TO BE PRINTED
C
C   GATHER INPUT DATA......
       DIMENSION PR(251),AUTOC(251)
       WRITE(6,1)
1      FORMAT('  EXPONENTIAL GAMMA / AUTOCORRELATION')
       PRINT  299
299    FORMAT(' ENTER 0 FOR GAMMA, 1 FOR AUTOCORRELATION')
       READ(5,298) IOP
298    FORMAT(I1)
       IF(IOP.EQ.1)PRINT 300,(I,I=1,10)
300    FORMAT(1H1,'      EXPECTED AUTOCORRELATIONS FOR EXPONENTIAL DELAYS'
     * ,//,23X'LAG:',/,'    MEAN',10I5)
C
C   PERFORM CALCULATIONS FOR DIFFERENT VALUES OF 'MEAN'
       DO 200 MEAN=1,20
       GAMMA=0.
C   WE NEED TO FILL UP THE PR VECTOR WITH PROBABILITIES
C   THIS WILL BE DONE THROUGH USE OF THE CUMULATIVE DENSITY
C   FUNCTION FOR THE EXPONENTIAL DISTRIBUTION
C   FN IS THE VALUE OF THE CUMULATIVE AT THE POINT OF INTEREST
C   F0 IS THE PREVIOUSLY ACCESSED VALUE OF THE CDF
C   THUS THE PROBABILITY IS FN-F0
C   SP WILL WILL THE SUM OF THE PR VECTOR.. USED TO CHECK
C   THE ACCURACY OF PR GENERATION
       F0=0
       LEN=6*MEAN
       SP=0.
C   INDEX IX INDICATES WHICH 'SLOT' WE ARE WORKING WITH
       DO 100 IX=1,LEN
       FN=1.-EXP(-FLOAT(IX)/MEAN)
       P=FN-F0
       SP=SP+P
       F0=FN
       PR(IX)=P
       GAMMA=GAMMA+P*(1.-P)
100     CONTINUE
       IF(SP.LT.0.99) PRINT 117,SP
117    FORMAT('  PROBABLE ERROR- SUM OF PROBABILITIES IS',F6.3)
C
C   WE CAN NOW GENERATE AUTOCOVARIANCES
C   INDEX I INDICATES THE SEPARATION LAG
C   INDEX J POINTS TO COMPONENTS SHARED BY PULSES OF LAG I
       LEN1=LEN-1
       DO 45 I=1,LEN1
       COV=0.
       MI=LEN-I
```

```
      DO 40 J=1,MI
40    COV=COV+PR(J)*PR(J+I)
45    AUTOC(I)=-COV/GAMMA
      IF(IOP.EQ.0)PRINT 110,MEAN,GAMMA
110   FORMAT(' MEAN=',I5,'  VAR/N=',F6.2)
C  FOLLOWING LOGIC  STOPS THE PRINTING OF ZEROS
      IP=10
      DO 301 I=1,10
      IF(AUTOC(I).LT.-0.005) GOTO 301
      IP=I
      GO TO 302
301   CONTINUE
302   IF(IOP.EQ.1)PRINT 305,MEAN,(AUTOC(I),I=1,IP)
305   FORMAT(3X,I2,3X,10F5.2)
C
C  GO TO NEXT VALUE OF MEAN
200   CONTINUE
      STOP
      END

.UNI
```

```
:D-H*TK2.ERL
C    THIS PROGRAM CALCULATES GAMMA AND
C   AUTOCORRELATION FOR CONSTANT INPUT INTO ERLANG3
C   DELAY
C   PR(I) IS THE PROBABILITY OF SELECTING A LAG OF I FTU'S
C   AUTOC(K) IS THE AUTOCORRELATION FOR SEPARATION K
C   MEAN IS THE EXPECTED THROUGHPUT TIME
C   LEN IS THE LENGTH  (OR NUMBER OF SLOTS) OF THE DELAY
C   XLAM IS 1./MEAN
C       DISTRIBUTION       ===   6 TIMES THE MEAN
C   IOP=1 CAUSES AUTOCORRELATION TO BE PRINTED
C   IOP=0 CAUSES GAMMA TO BE PRINTED
C
C   GATHER INPUT DATA......
        DIMENSION PR(251),AUTOC(251)
        WRITE(6,1)
1       FORMAT('  ERLANG3 GAMMA / AUTOCORRELATION')
        PRINT  299
299     FORMAT(' ENTER 0 FOR GAMMA, 1 FOR AUTOCORRELATION')
        READ(5,298) IOP
298     FORMAT(I1)
        IF(IOP.EQ.1)PRINT 300,(I,I=1,10)
300     FORMAT(1H1,'      EXPECTED AUTOCORRELATIONS FOR ERLANG3 DELAYS'
      * ,//,23X'LAG:',/,'    MEAN',10I5)
C
C   PERFORM CALCULATIONS FOR DIFFERENT VALUES OF 'MEAN'
        DO 200 MEAN=1,20
C   WE NEED TO FILL UP THE PR VECTOR WITH PROBABILITIES
C   THIS WILL BE DONE THROUGH A CALL TO SUBROUTINE 'FILL'
        LEN=6*MEAN
        XLAM=1./FLOAT(MEAN)
C   CREATE GAMMA AND SP (SP IS THE SUM OF ALL PR'S TO
C   CHECK PR CALCULATIONS)
        GAMMA=0.
        SP=0.
        CALL FILL(LEN,XLAM,PR)
        DO 100 IX=1,LEN
        P=PR(IX)
        SP=SP+P
        GAMMA=GAMMA+P*(1.-P)
100      CONTINUE
        IF(SP.LT.0.99) PRINT 117,SP
117     FORMAT('  PROBABLE ERROR- SUM OF PROBABILITIES IS',F6.3)
C
C   WE CAN NOW GENERATE AUTOCOVARIANCES
C   INDEX I INDICATES THE SEPARATION LAG
C   INDEX J POINTS TO COMPONENTS SHARED BY PULSES ON LAG I
        LEN1=LEN-1
        DO 45 I=1,LEN1
        COV=0.
        MI=LEN-I
        DO 40 J=1,MI
40      COV=COV+PR(J)*PR(J+I)
45      AUTOC(I)=-COV/GAMMA
        IF(IOP.EQ.0)PRINT 110,MEAN,GAMMA
110     FORMAT(' MEAN=',I5,'  VAR/N=',F6.2)
C  FOLLOWING LOGIC  STOPS THE PRINTING OF ZEROS
```

```
      IP=10
      DO 301 I=1,10
      IF(AUTOC(I).LT.-0.005) GOTO 301
      IP=I
      GO TO 302
301   CONTINUE
302   IF(IOP.EQ.1)PRINT 305,MEAN,(AUTOC(I),I=1,IP)
305   FORMAT(3X,I2,3X,10F5.2)
C
C  GO TO NEXT VALUE OF MEAN
200   CONTINUE
      STOP
C
      SUBROUTINE FILL(LEN,XLAM,PR)
      DIMENSION PR(1)
C  LEN IS THE NUMBER OF INTERVALS TO CALCULATE PROBABILITIES
C  PR(I) IS THE PROBABILITY OF SELECTING A LAG OF I FTU'S
C     IT IS CALCULATED BY INTEGRATION UNDER THE P.D.F. OF THE
C     ERLANG3 DISTRIBUTION
C  XLAM IS 1/MEAN
C  N IS THE NUMBER OF SUBINTERVALS TO USED MIDPOINT
C     METHOD OF APPROXIMATE INTEGRATION
      N=6
C  XL3 AND XXX ARE TEMPORARY VARIABLES TO HOLD COMPONENTS
C   OF THE ERLANG3 PROB.DENSITY.FUNC.
      XL3=XLAM*3.
      XXX=13.5*XLAM**3
      DELTA=1./FLOAT(N)
      TIME=DELTA/2.
C  CALCULATE PROBABILITY FOR EACH OF THE 'LEN' SEGMENTS
C  INDEX I POINTS TO THE INTERVAL OF INTEREST
C  INDEX J POINTS TO THE SUBINTERVAL OF INTERVAL I
      DO 100 I=1,LEN
      AREA=0.
C  CALCULATE AREA UNDER CURVE VIA MIDPOINT ALGORITHM, N SECTIONS
      DO 50 J=1,N
C  GET FUNCTION VALUE AT THIS POINT 'TIME'
      FUNCV=XXX*EXP(-XL3*TIME)*TIME**2
      AREA=AREA+FUNCV*DELTA
      TIME=TIME+DELTA
50    CONTINUE
      PR(I)=AREA
100   CONTINUE
      RETURN
      END

.EXP
```

APPENDIX II

PROGRAMS TO CALCULATE THE ATTENUATION OF SIGNALS PASSING THROUGH A NORMAL OR ERLANG3 DELAY

This appendix contains two programs which calculate attenuation: one for normal delays, and one for Erlang3 delays. The programs create sinusoidal input in order to calculate the expected value and variance of each pulse of an output wavelength. Attenuation is then calculated using the expected output series and equation 4-10.

Program Logic:

1. Read average input, signal amplitude, signal wavelength and FTU.
2. Create FTU adjusted variables.
3. Generate input series XIN(.) by equation 4-8.
4. Determine input magnitude from XIN(.).
5. Define delay width; iterate through #11.
6. Create E(.) array as delay distribution.
7. Use equation 4-6 to create EXPT(.), expected value of output.
8. Use equation 4-7 to create VAR(.), variance of output pulses.
9. From EXPT(.), calculate output magnitude, then attenuation as the ratio of output magnitude over input magnitude.
10. Output attenuation, expected values, and variance of output values.
11. Go to #5 to perform calculation on wider delay distribution.

```
ID-H*TV.NOR1
    C  THIS ROUTINE CALCULATES THE VARIANCE (TIME DEPENDENT)
    C  OF A SINUSOID SIGNAL ENTERING A RANDOM (NORMAL)
    C  DELAY BOX
          DIMENSION VPROB(151),PROB(151)
          INTEGER AMPLI
          REAL INMAG
    C
    C  PROB(I) IS THE PROBABILITY OF SELECTING A LAG OF I FTU'S
    C  VPROB(I) IS PROB(I)*(1-PROB(I))   USED IN VARIANCE CALCUL.
    C  XIN(I) IS THE INPUT VECTOR
    C  EXPT(J) IS THE EXPECTED VALUE OF OUTPUT PULSE J
    C  VAR(J) IS THE VARIANCE OF OUTPUT PULSE J
    C  SD(J) IS THE STANDARD DEVIATION OF OUTPUT PULSE J
    C
    C  IDATA DETERMINE WHETHER TO PRINT ALL, OR SOME
    C  LEVEL IS THE CARRIER HEIGHT
    C  AMPLI IS THE AMPLITUDE OF THE INPUT SIGNAL
    C  INMAG IS THE MAGNITUDE OF THE INPUT SIGNAL
    C  OUTMAG IS THE MAGNITUDE OF THE OUTPUT SIGNAL
    C  WAVEL IS THE WAVELENGTH(PERIOD) OF THE INPUT SIGNAL
    C  SIGMA IS THE STANDARD DEVIATION OF THE DELAY
          INTEGER WAVEL,SIGMA
          DIMENSION XIN(100),VAR(100),EXPT(100),SD(100)
          PRINT 1
    1     FORMAT(' NORMAL+++ SHOULD FULL DATA BE GIVEN? (0=NO,1=YES)')
          READ(5,9) IDATA
    2     PRINT 4
    4     FORMAT(' PLEASE ENTER LEVEL,AMPLITUDE,WAVELENGTH IN FF (INTEGER)')
          READ(5,9) LEVEL,AMPLI,WAVEL
    9     FORMAT()
          IF(LEVEL.LE.0) STOP
          IF(LEVEL.LT.AMPLI) GO TO 2
          PRINT 10,LEVEL,AMPLI,WAVEL
    10    FORMAT(' LEVEL=',I4,'  AMPLITUDE=',I4,' WAVELENGTH=',I4)
    C
          IF(WAVEL.LT.100) GO TO 20
          PRINT 19
    19    FORMAT('   WAVELENGTH TOO LONG****')
          GO TO 2
    C GENERATE INFLOW VALUES
    20    INLEN = 150+WAVEL
          XWAVE=WAVEL
          OMEGA=6.28318/XWAVE
          XLEVEL=LEVEL
          DO 200 I=1,INLEN
    200   XIN(I)=AMPLI*SIN(OMEGA*I)+XLEVEL
    C  DETERMINE INPUT MAGNITUDE
          IW2=WAVEL/2
          INMAG=0
          DO 205 I=1,IW2
    205   INMAG=INMAG+(XIN(I)-XLEVEL)
          IF(IDATA.GT.0) PRINT 207,INMAG,(XIN(I),I=1,WAVEL)
    207   FORMAT(' INMAG=' F8.2,3X,'INPUT VALUES ARE:',/,(5F7.2))
    C
    C  ITERATE ON LEVEL DELAYS OF 1 THRU 10
          DO 1000 SIGMA=1,30
```

```
      LEN=6*SIGMA
      IMEAN=3*SIGMA
C  FILL UP THE PROB AND VPROB ARRAYS:::::
      SP=0.
      F0=0.
      DO 100 K=1,LEN
      FN=ZVAL(FLOAT(K-IMEAN)/FLOAT(SIGMA))
      P=FN-F0
      F0=FN
      PROB(K)=P
      SP=SP+P
      VPROB(K)=P*(1.-P)
100   CONTINUE
      IF(IDATA.EQ.1) PRINT 105,(PROB(I),I=1,LEN)
105   FORMAT('0   PROBABILITIES:',/,(5F8.3))
      IF(SP.LT.0.99) PRINT 110,SP
110   FORMAT('  PROBABLE ERROR, SUM OF PROBABILITIES IS' F6.3)
C
C  GENERATE ONE PERIOD IN THE OUTPUT SERIES
C  INDEX IJK IS THE POSITION OF THE OUTPUT SERIES
C  INDEX IV IS THE CONVOLUTION POINTER
C
      DO 400 IJK=1,WAVEL
      VARX=0.
      EXPX=0.
      DO 300 IV=1,LEN
      VARX=VARX+XIN(IJK+IV)*VPROB(IV)
      EXPX=EXPX+XIN(IJK+IV)*PROB(IV)
300   CONTINUE
      VAR(IJK)=VARX
      EXPT(IJK)=EXPX
      SD(IJK)=SQRT(VARX)
400   CONTINUE
      OUTMAG=0.
      XMAX=XLEVEL
      DO 820 I=1,WAVEL
      IF(EXPT(I).GT.XMAX) XMAX=EXPT(I)
820   IF(EXPT(I).GT.XLEVEL) OUTMAG=OUTMAG+(EXPT(I)-XLEVEL)
      ATTEN=OUTMAG/INMAG
      PRINT 830, SIGMA,XMAX,ATTEN
830   FORMAT(' SIGMA=',I3,' MAXIMUM VALUE=' F6.1,' ATEN=' F6.2)
      IF(IDATA.EQ.0) GO TO 870
      PRINT 850, (EXPT(IK),VAR(IK),SD(IK),IK=1,WAVEL)
850   FORMAT('     EXPT      VAR     S.D.',/,(3F8.2))
      PRINT 851
851   FORMAT(/)
870   IF(ATTEN.LT.0.01) GO TO 2
1000  CONTINUE
      GO TO 2
      FUNCTION ZVAL(X)
      INTEGER HIGH
C  DETERMINE CUMULATIVE PROBABILITY OF N(0,1) AT THE
C  POINT X
C  VECTOR VAL HOLDS CUMULATIVE VALUES FOR THE NORMAL
C  IN TERMS OF .1 SIGMAS....     VAL(1) IS THE CUMULATIVE
C  AT -.0 SIGMA        VAL(I) IS THE CUMULATIVE AT
C  THE POINT -(I-1)/10 SIGMA
```

```
      DIMENSION VAL(31)
      DATA VAL/.5000,.4602,.4207,.3821,.3446,
     *   .3085,.2743,.2420,.2119,.1841,
     *   .1587,.1357,.1151,.0968,.0808,
     *   .0668,.0548,.0446,.0359,.0287,
     *   .0228,.0179,.0139,.0107,.0082,
     *   .00621,.00466,.00347,.00256,.00187,.000968/
C  ASSUME 'X' WAS NEGATIVE
      Y=ABS(X)*10.+1
C  NOTE THAT THE *10 CHANGES METRIC TO .1 SIGMAS AND THE
C  +1 SHIFTS ORIGIN SINCE VAL(1) IS FOR A .0 SIGMA VALUE
C  THE CORRECT VALUE FOR Y LIES BETWEEN LOW AND HIGH
      LOW=Y+1
      HIGH=Y
      ZVAL=0.
      IF(LOW.LE.30) ZVAL=VAL(LOW)+(LOW-Y)*(VAL(HIGH)-VAL(LOW))
C  IF 'X' WAS NOT NEGATIVE, CORRECT ZVAL VALUE
      IF(X.GT.0.) ZVAL=1.-ZVAL
      RETURN
      END
```

.SIM4

```
ID-H*TV.FRL
      C  THIS ROUTINE CALCULATES THE VARIANCE (TIME DEPENDENT)
      C  OF A SINUSOID SIGNAL ENTERING A RANDOM (ERLANG)
      C  DELAY BOX
            DIMENSION VPROB(151),PROB(151)
            INTEGER AMPLI
            REAL LAMBDA,INMAG
      C
      C  PROB(I) IS THE PROBABILITY OF SELECTING A LAG OF I FTU'S
      C  VPROB(I) IS PROB(I)*(1-PROB(I))   USED IN VARIANCE CALCUL.
      C  XIN(I) IS THE INPUT VECTOR
      C  EXPT(J) IS THE EXPECTED VALUE OF OUTPUT PULSE J
      C  VAR(J) IS THE VARIANCE OF OUTPUT PULSE J
      C  SD(J) IS THE STANDARD DEVIATION OF OUTPUT PULSE J
      C
      C  IDATA DETERMINE WHETHER TO PRINT ALL, OR SOME
      C  LEVEL IS THE CARRIER HEIGHT
      C  AMPLI IS THE AMPLITUDE OF THE INPUT SIGNAL
      C  INMAG IS THE MAGNITUDE OF THE INPUT SIGNAL
      C  OUTMAG IS THE MAGNITUDE OF THE OUTPUT SIGNAL
      C  WAVEL IS THE WAVELENGTH(PERIOD) OF THE INPUT SIGNAL
      C  MEAN IS THE EXPECTED THROUGHPUT TIME OF THE DELAY
            INTEGER WAVEL
            DIMENSION XIN(100),VAR(100),EXPT(100),SD(100)
            PRINT 1
      1     FORMAT(' ERLANG+++ SHOULD FULL DATA BE GIVEN? (0=NO,1=YES)')
            READ(5,9) IDATA
      2     PRINT 4
      4     FORMAT(' PLEASE ENTER LEVEL,AMPLITUDE,WAVELENGTH IN FF (INTEGER)')
            READ(5,9) LEVEL,AMPLI,WAVEL
      9     FORMAT()
            IF(LEVEL.LE.0) STOP
            IF(LEVEL.LT.AMPLI) GO TO 2
            PRINT 10,LEVEL,AMPLI,WAVEL
      10    FORMAT(' LEVEL=',I4,'  AMPLITUDE=',I4,' WAVELENGTH=',I4)
      C
            IF(WAVEL.LT.100) GO TO 20
            PRINT 19
      19    FORMAT('   WAVELENGTH TOO LONG****')
            GO TO 2
      C GENERATE INFLOW VALUES
      20    INLEN = 150+WAVEL
            XWAVE=WAVEL
            OMEGA=6.28318/XWAVE
            XLEVEL=LEVEL
            DO 200 I=1,INLEN
      200   XIN(I)=AMPLI*SIN(OMEGA*I)+XLEVEL
      C  DETERMINE INPUT MAGNITUDE
            IW2=WAVEL/2
            INMAG=0
            DO 205 I=1,IW2
      205   INMAG=INMAG+(XIN(I)-XLEVEL)
            IF(IDATA.GT.0) PRINT 207,INMAG,(XIN(I),I=1,WAVEL)
      207   FORMAT(' INMAG=' F8.2,3X,'INPUT VALUES ARE:',/,(5F7.2))
      C
      C  ITERATE ON LEVEL DELAYS OF 1 THRU 10
            DO 1000 MEAN=1,30
```

```
      LEN=5*MEAN
      LAMBDA=1./FLOAT(MEAN)
C  FILL UP THE PROB AND VPROB ARRAYS:::::
      CALL FILL(LEN,LAMBDA,PROB)
      IF(IDATA.EQ.1) PRINT 45,(PROB(I),I=1,LEN)
45    FORMAT(/'  PROBABILITIES:',/,(5F8.3))
      SP=0.
      DO 100 K=1,LEN
      P=PROB(K)
      SP=SP+P
      VPROB(K)=P*(1.-P)
100   CONTINUE
      IF(SP.LT.0.99) PRINT 110,SP
110   FORMAT('  PROBABLE ERROR, SUM OF PROBABILITIES IS' F6.3)
C
C  GENERATE ONE PERIOD IN THE OUTPUT SERIES
C  INDEX IJK IS THE POSITION OF THE OUTPUT SERIES
C  INDEX IV IS THE CONVOLUTION POINTER
C
      DO 400 IJK=1,WAVEL
      VARX=0.
      EXPX=0.
      DO 300 IV=1,LEN
      VARX=VARX+XIN(IJK+IV)*VPROB(IV)
      EXPX=EXPX+XIN(IJK+IV)*PROB(IV)
300   CONTINUE
      VAR(IJK)=VARX
      EXPT(IJK)=EXPX
      SD(IJK)=SQRT(VARX)
400   CONTINUE
      OUTMAG=0.
      XMAX=XLEVEL
      DO 820 I=1,WAVEL
      IF(EXPT(I).GT.XMAX) XMAX=EXPT(I)
820   IF(EXPT(I).GT.XLEVEL) OUTMAG=OUTMAG+(EXPT(I)-XLEVEL)
      ATTEN=OUTMAG/INMAG
      PRINT 830, MEAN,XMAX,ATTEN
830   FORMAT(' MEAN=',I3,' MAXIMUM VALUE=' F6.1,' ATEN=' F6.2)
      IF(IDATA.EQ.0) GO TO 870
      PRINT 850, (EXPT(IK),VAR(IK),SD(IK),IK=1,WAVEL)
850   FORMAT('     EXPT      VAR     S.D.',/,(3F8.2))
      PRINT 851
851   FORMAT(/)
870   IF(ATTEN.LT.0.01) GO TO 2
1000  CONTINUE
      GO TO 2
C
      SUBROUTINE FILL(LEN,XLAM,PR)
      DIMENSION PR(1)
C  LEN IS THE NUMBER OF INTERVALS TO CALCULATE PROBABILITIES
C  PR(I) IS THE PROBABILITY OF SELECTING A LAG OF I FTU'S
C     IT IS CALCULATED BY INTEGRATION UNDER THE P.D.F. OF THE
C     ERLANG3 DISTRIBUTION
C  XLAM IS 1/MEAN
C  N IS THE NUMBER OF SUBINTERVALS TO USED MIDPOINT
C     METHOD OF APPROXIMATE INTEGRATION
      N=6
```

```
C   XL3 AND XXX ARE TEMPORARY VARIABLES TO HOLD COMPONENTS
C    OF THE ERLANG3 PROB.DENSITY.FUNC.
       XL3=XLAM*3.
       XXX=13.5*XLAM**3
       DELTA=1./FLOAT(N)
       TIME=DELTA/2.
C   CALCULATE PROBABILITY FOR EACH OF THE 'LEN' SEGMENTS
C   INDEX I POINTS TO THE INTERVAL OF INTEREST
C   INDEX J POINTS TO THE SUBINTERVAL OF INTERVAL I
       DO 100 I=1,LEN
       AREA=0.
C   CALCULATE AREA UNDER CURVE VIA MIDPOINT ALGORITHM, N SECTIONS
       DO 50 J=1,N
C   GET FUNCTION VALUE AT THIS POINT 'TIME'
       FUNCV=XXX*EXP(-XL3*TIME)*TIME**2
       AREA=AREA+FUNCV*DELTA
       TIME=TIME+DELTA
50     CONTINUE
       PR(I)=AREA
100    CONTINUE
       RETURN
       END
```

.NOR1

APPENDIX III

DISCRETE FORMULATION OF INVENTORY SYSTEM

This appendix contains a Fortran implementation of the base discrete inventory model, model IS. The variance of the noise applied to the sales is input controlled, and thus model I (deterministic sales) is mimed by model IS using zero sales variance. The program automatically performs the requested simulation using 10 internally defined random number seeds. The function RANDOM used to create random [uniform on (0.,1.0] values is machine-dependent, and is currently defined to operate on the Univac 1108 or other 36 bit machine.

Array EXIT(.) acts as the accumulation file. EXIT(.) is maintained as a sorted file through the use of a parallel pointer array IP(.). LEAVE, NEXT and END point to the first-to-leave, next-free, and last-free locations of array EXIT which maintains two subfiles: an in-use subfile and free (i.e., not in-use) subfile.

Program Logic:

1. Read initial values and sales variance.
2. Repeat through #12 for each seed value.
    3. Initialize system and EXIT(.) and IP(.) sarrays.
    4. Repeat through #10 for each time period.
        5. For each newly requested unit, draw production duration time and store unit in EXIT(.), maintaining IP(.) as sorting pointers.
        6. Check to see if any units exit production (the EXIT array) at this time. If so move them to inventory via removing them from EXIT and

and incrementing INVEN variable.

7. Create draw for day's sales.
8. Withdraw sold units from inventory.
9. Calculate number of units to order.
10. Go to #4 for next time period.

11. Output Values at end of simulation run.

12. Go to #2 for next seed value to use for simulation.

```
ID-H*TK.SIM4A
C
C  THIS ROUTINE WILL SIMULATE THE INVENTORY SYSTEM
C  USING TEN DIFFERENT SEEDS STORED IN VECTOR ISD
C  THE MODELING APPROACH IS ITEM BY ITEM
C  THE VECTORS EXIT(I) AND IP(I) ARE USED IN THE FILE
C  STRUCTURE OF THE MODEL.  EXIT(I) IS THE EXIT TIME FOR
C  ENTITY I   (WILL BE 999999 IF LOCATION IS EMPTY)
C
C  LEAVE POINTS TO THE FIRST MEMBER OF EXIT TO LEAVE
C  NEXT POINTS TO THE NEXT FREE LOCATION IN VECTOR EXIT
C  END POINTS TO THE LAST FREE LOCATION OF VECTOR EXIT
C
C  THUS THE 'FREE' FILE AND 'USED' FILE ARE IN THE EXIT VECTOR
C  THE 'USED' FILE IS KEPT IN SORTED ORDER, AS A FUNCTION
C  OF THE EXIT TIME.
C  IF SLOT J IS A MEMFBER OF THE USED FILE, IP(J) POINTS
C  TO THE NEXT SLOT OF THE USED FILE TO EXIT
C  IF SLOT J IS A MEMBER OF THE FREE FILE, IP(J) POINTS TO
C  ANOTHER MEMBER OF THE FREE FILE
C
      DIMENSION ISD(10)
      DATA ISD/12344321,31415962,35355311,357911,4361,
     * 31159,30777713,12233457,20394857,53537511/
      COMMON /T/ITIME
      INTEGER EXIT (1000),IP(1000),OUT,END
      INTEGER OUTFLO,SALES
C
C  SINCE MULTIPLE RUNS ARE TO BE MADE, INITIAL MODEL
C  VALUES ARE STORED IN VARIABLES INA, INVENA
C  INA IS THE STARTING NUMBER OF ENTITIES IN THE DELAY
C  INVENA IS THE NUMBER IN THE INVENTORY
C  IN WILL BE USED AS THE CURRENT NUMBER IN THE DELAY
C  INVEN WILL BE USED AS THE CURRENT NUMBER IN INVENTORY
C  DELAY IS THE MEAN THROUGHPUT TIME
C  DESIRE IS THE DESIRED INVENTORY LEVEL
C  REACT IS THE REACTION RATE, OR ADJUSTMENT TIME
C  SVAR IS THE VARIANCE OF SALES
C  SIG IS THE STANDARD DEVIATION OF SALES
C  ID IS THE OUTPUT CYCLE, IE HOW MANY DT'S PER PRINTING
      PRINT 101
101   FORMAT(' PLEASE ENTER STARTING-LEVEL AND INVENTORY IN FF',
     * ' (INTEGERS)')
      READ (5,102) INA,INVENA
102   FORMAT()
      PRINT 103,INA,INVENA
103   FORMAT(' LEVEL=' I3,'  INVENTORY=',I3)
      PRINT 104
104   FORMAT(' PLEASE ENTER MEAN-DELAY-TIME AND DESIRED INV',
     * 'ENTORY (REALS)')
      READ(5,102) DELAY,DESIRE
      PRINT 105,DELAY,DESIRE
105   FORMAT(' DELAY='F6.1,'  DESIRED-INV=' F6.1)
      PRINT 106
106   FORMAT(' PLEASE ENTER  DT AND REACT-RATE (REALS)')
      READ(5,102) DT,REACT
      PRINT 107,DT,REACT
```

```
107    FORMAT(' DT=' F5.2,'  REACT-RATE=' F5.1)
       PRINT 3977
3977   FORMAT('  PLEASE ENTER THE VARIANCE OF SALES')
       READ(5,102) SVAR
       SIG=SQRT(SVAR)
       PRINT 3978,SVAR
3978   FORMAT('  VARIANCE OF SALE=',F6.2)
       PRINT 108
108    FORMAT(' PLEASE ENTER THE DATA-CYCLE  (INTEGER)')
       READ(5,102) ID
110    FORMAT(///,' CYCLE   TIME   IN  LEV  OUT  INV  SALES')
C
C  PERFORM SIMULATIONS FOR THE 10 DIFFERNET SEEDS
       DO 3989 IUYT=1,10
       ISEED=ISD(IUYT)
       LEVEL=0.
       WRITE(6,3988) ISEED
3988   FORMAT(1H1,'  SEED IS:',I9)
       PRINT 110
C  INITIALIZE VARIABLES TO STARTING VALUES
       IN=INA
       INVEN=INVENA
       ITIME=0
       DELAYZ=DELAY/DT
       TIME=0.
C  EMPTY OUT THE EXIT ARRAY AND SET POINTERS CORRECTLY
C  NOTE LOCATION 1 IS CONSIDER FULL AT ALL TIMES, WITH
C  AN ENTITY TO LEAVE AT TIME 999999
       DO 2 I=1,1000
       EXIT(I)=999999
2      IP(I)=I+1
       NEXT=2
       LEAVE=1
       END=1000
C
C  XM IS USED AS A SUMMATION OF INVENTORY VALUES
C  XV IS USED AS A SUMMATION OF INVENTORY**2 VALUES
C  XM ANS XV WILL BE USED TO CALCULATE MEAN AND VARIANCE
C  OF INVENTORY
C  IN IS THE NUMBER OF ENTITIES TO ENTER THE DELAY
C  TIME IS THE CURRENT TIME VALUE
C  ITERATE THROUGH SYSTEM
       XM=0.
       XV=0.
       MAX=150./DT
       DO 1789 ITIME=1,MAX
       IF(IN.EQ.0) GO TO 40
       DO 10 I=1,IN
           IX=ITIME+.5-DELAYZ*ALOG(RANDOM(ISEED))
       CALL ENTER(IX,EXIT,IP,NEXT,LEAVE)
10     CONTINUE
C
C  DETERMINE NUMBER TO LEAVE FROM BOX
40     OUT=OUTFLO(EXIT,IP,NEXT,LEAVE,END)
       TIME=TIME+DT
       LEVEL=LEVEL+IN-OUT
C  GENREATE SALES VALUE
```

```
      XSS=20.*DT+XNOR(ISEED,SIG)
      SALES=XSS
      IF(RANDOM(ISEED).LT.(XSS-FLOAT(SALES)))SALES=SALES+1
      SALES=MAX0(0,MIN0(INVEN,SALES))
      INVEN=INVEN+OUT-SALES
C
C  OUTPUT VALUES
      IG=MOD(ITIME,ID)
      IF(IG.EQ.0)PRINT 77,ITIME,TIME,IN,LEVEL,OUT,INVEN,SALES
77     FORMAT(I5,F8.2,5I5)
      XM=XM+INVEN
      XV=XV+(INVEN-DESIRE)**2
C
C  CALCULATE NUMBER TO ENTER
C  FRACTIUANL PARTS GETS A RANDOM  DRAW OF SAID PROBABILITY
      XIN=(DESIRE-FLOAT(INVEN))/REACT*DT+SALES
      IN=XIN
      RES=XIN-FLOAT(IN)
      IF(RANDOM(ISEED).LT.RES) IN=IN+1
      IN=MAX0(IN,0)
1789  CONTINUE
      XM=XM/FLOAT(MAX)
      XV=XV/FLOAT(MAX)
      PRINT 1790,XM,XV
1790  FORMAT(' INVENTORY MEAN:',F6.2,'   VARIANCE:',F8.2)
C
C GO AND REDO SIMULATION WITH ANOTHER SEED
3989  CONTINUE
      STOP
C
C    ======::::::::======:::::::::::==========
C       SUBROUTINES FOLLOW
C
      SUBROUTINE ENTER(IX,EXIT,IP,FREE1,USED1)
C  A ROUTINE TO ENTER 'IX' INTO THE ARRAY 'EXIT'
C  'IP' IS A POINTER ARRAY SHOWING THE SORTED ORDER OF 'EXIT'
C  'FREE1' POINTS TO THE FIRST MEMBER OF THE FREE SUBFILE
C  'END' POINTS TO THE LAST MEMBER OF THE FREE SUBFILE
C  'USED1' POINTS TO THE FIRST MEMBER OF THE USED SUBFILE.
      INTEGER EXIT(1),IP(1),USED1,FREE1
      EXIT(FREE1)=IX
      IT=USED1
      IF(IX.LE.EXIT(IT)) GO TO 8
7     ITI=IT
      IT=IP(IT)
      IF(IX.GT.EXIT(IT)) GO TO 7
      IP(ITI)=FREE1
      NN=FREE1
      FREE1=IP(FREE1)
      IP(NN)=IT
      RETURN
8     USED1=FREE1
      FREE1=IP(FREE1)
      IP(USED1)=IT
      RETURN
C
      INTEGER FUNCTION OUTFLO(EXIT,IP,FREE1,USED1,END)
```

```
C  THIS ROUTINE SEARCHES THE EXIT ARRAY FOR ANY MEMBERS
C     WISHING TO LEAVE AT 'ITIME'.  THE SLOTS ARE THEN
C     ATTACHED TO THE FREE SUBFILE.
      INTEGER EXIT(1),IP(1),OUT,USED1,FREE1,END
      COMMON /T/ ITIME
      OUT=0
50    IF(EXIT(USED1).GT.ITIME) GO TO 59
      OUT=OUT+1
C  VACATE SLOT AND TACK SLOT ONTO EMPTY FILE
      EXIT(USED1)=9999999
      IP(END)=USED1
      END=USED1
      USED1=IP(USED1)
      GO TO 50
59    OUTFLO=OUT
      RETURN
C
C
      FUNCTION RANDOM(I)
C  THIS ROUTINE GENERATES UNIFORM (0.-1.) RANDOM DEVIATES.
      I=I*131075
      IF(I.LE.0)I=-I
      RANDOM=I*.2910383E-10
      RETURN
      FUNCTION XNOR(IRN,STD)
      DATA C1,C2,C3,C4,C5,C6/2.515517,.802853,
     *.010328,1.432788,.189269,.001308/
      URN=RANDOM(IRN)
C          RANDOM IS THE UNIFORM DEVIATE GENERATOR
C          XNOR IS THE NORMAL DEVIATE GENERATOR
C          IRN = THE RANDOM NUMBER SEED
C          STD = THE STANDARD DEVIATION OF THE NORMAL DISTRIBUTION
      ARG=.5*(1.-ABS(1.-2.*URN))
      V=SQRT(-2.*ALOG(ARG))
      PN= C1 + C2*V + C3*V**2
      PD= 1. + C4*V + C5*V**2 + C6*V**3
      A=(URN-.5)/ABS(URN-.5)
      B=V-(PN/PD)
      XNOR= A*STD*B
      RETURN
C
      END
```

THIS LISTING PRODUCED ON JUNE 15, 1975   AT 01:09:30

****   **** PLEASE RETURN THIS LISTING TO:

:AbDX

APPENDIX IV

CONTINUOUS FORMULATION OF INVENTORY SYSTEM

This appendix contains the listing for model CS+. This, like all the continuous inventory models, contains a state space exponential delay to represent the production delay, with the option for user-specified noise added to production output or to the daily sales or both. All other continuous models can be imitated by selective control of these noise sources. Model C is simulated via requesting zero variance for both noise sources; model CS is mimed via requesting zero production variance and positive sales variance; model C+ is represented with positive production variance and zero sales variance.

Program Logic:

1. Read initial values of system.
2. Read seed and variance for both production and sales noise sources.
3. Initialize system.
4. Cycle through #9 for each time period.
    5. Create production exits as sum of exponential predicted plus random draw.
    6. Create sales as sum of deterministic value plus random draw.
    7. Adjust production accumulation and inventory to reflect effect of #5 and #6.
    8. Collect data on daily variance
    9. Go to #4 for next DT time period.

10. Output values at end of simulation.

11. Go to #2 to accept new seeds and variances.

```
'ID-H*TK.SIM4
      C
      C   A PROGRAM TO SIMULATE THE INVENTORY SYSTEM USING
      C   CONTINUOUS SIMULATION
      C
      C           FOR EACH DT::::::::
      C   LEVEL IS THE NUMBER OF ENTITIES IN THE DELAY
      C  OUT IS THE NUMBER LEAVING THE DELAY
      C   IN IS THE NUMBER ENTERING THE DELAY
      C   INVEN IS THE CONTENTS OF INVENTORY
      C   SALES IS THE OUTFLOW FROM THE INVENTORY
      C
      C
      C   ID IS THE DATA CYCLE INDICATING THE NUMBER OF DT'S
      C      PER PRINTING OF SYSTEM STATUS
             COMMON LEVEL,OUT,IN,INVEN,MEAN,DT,SALES
             REAL MEAN
             REAL IN,OUT,LEVEL,INVEN
             REAL  LEVEL9,INVEN9
      C
      C   SET UP INITIAL VALUES AND PARAMETERS
             PRINT 1
      1      FORMAT(' PLEASE INPUT:  MEAN,REACT,DESIRE,LEVEL,INVEN,DT '
            *,'IN FF (REALS)')
      C
              READ(5,2) MEAN,REACT,DESIRE,LEVEL9,INVEN9,DT
      2      FORMAT()
      C   MEAN IS THE AVERAGE THROUGHPUT TIME
      C   REACT IS THE REACTION RATE OR ADJUSTMENT TIME
      C   DESIRE IS THE DESIRED INVENTORY LEVEL
      C   LEVEL9 IS THE INITIAL ACCUMULATIN  VALUE
      C   INVEN9 IS THE INITIAL INVENTORY
             PRINT 3,MEAN,REACT,DESIRE,LEVEL9,INVEN9,DT
      3      FORMAT(' MEAN='F4.1,'  REACT='F4.1,'  DESIRE=',
            * F4.1,/,' LEVEL=',F5.1,' INVENTORY=',F5.1,'  DT=' F4.2)
             PRINT 6
      6      FORMAT(' PLEASE ENTER DATA CYCLE (INTEGER)')
             READ(5,2) ID
      551    WRITE(6,698)
      698    FORMAT(' ENTER SEED AND VAR FOR DELAY NOISE, AND SALES NOISE')
      C   ISEED1 IS THE SEED TO BE USED WITH DELAY NOISE
      C   DVAR IS THE VARIANCE OF DELAY NOISE
      C   DSIG IS THE SIGMA OF DELAY NOISE
      C   ISEED2 IS THE SEED TO BE USED WITH SALES NOISE
      C   SVAR IS THE VARIANCE OF SALES NOISE
      C   SSIG IS THE SIGMA OF SALES NOISE
             READ(5,2) ISEED1,DVAR,ISEED2,SVAR
             DSIG=SQRT(DVAR)
             IF(ISEED1.EQ.0) STOP
             SSIG=SQRT(SVAR)
             WRITE(6,699) ISEED1,DVAR,ISEED2,SVAR
      699    FORMAT(' DELAY SEED,VAR:' I9,F6.2,'      SALES SEED,VAR:'I9,F6.2)
             PRINT 7
      7      FORMAT(///,' CYCLE   TIME       IN   LEVEL      OUT   INVEN',
            * ,'    SALES')
      C
      C   INITIALIZE SYSTEM TO STARTING CONDITIONS
```

```
      LEVEL=LEVEL9
      INVEN=INVEN9
      TIME=0.
      IN=0.
C   XS AND XV ARE SUMMATIONS USED TO GET MEAN AND VARIANCE
      XS=0
      MAX=150./DT
      XV=0.
C
C   CYCLE THROUGH SYSTEM
      DO 100 IDT=1,MAX
      TIME=TIME+DT
C
C   CREATE OUTFLOW AND SALES
      OUT=LEVEL/MEAN*DT+XNOR(ISEED1,DSIG)
      OUT=AMAX1(0.,AMIN1(LEVEL,OUT))
      SALES=20.*DT+XNOR(ISEED2,SSIG)
      SALES=AMAX1(0.,AMIN1(INVEN,SALES))
C
      LEVEL=LEVEL+IN-OUT
      INVEN=INVEN+OUT-SALES
      IN=(DESIRE-INVEN)/REACT*DT+SALES
      IN=AMAX1(IN,0.)
C   SEE IF IT IS TIME TO PRINT
      IG=MOD(IDT,ID)
      IF(IG.EQ.0)PRINT 10, IDT,TIME,IN,LEVEL,OUT,INVEN,SALES
10    FORMAT(I5,6F8.1)
C   XS AND XV ARE SUMMATIONS USED TO CALCULATE DELAY
C    AND VARIANCE OF INVENTORY
      XS=XS+INVEN
      XV=XV+(INVEN-DESIRE)**2
100   CONTINUE
      XM=XS/FLOAT(MAX)
      XV=XV/FLOAT(MAX)
      PRINT 101,XM,XV
101   FORMAT(' DELAY INVENTORY:',F6.2,'  VARIANCE:',F8.1)
      GO TO 551
C
      FUNCTION XNOR(IRN,SIG)
      DATA C1,C2,C3,C4,C5,C6/2.515517,.802853,
     *.010328,1.432788,.189269,.001308/
      URN=RANDOM(IRN)
C         RANDOM IS THE UNIFORM DEVIATE GENERATOR
C         XNOR IS THE NORMAL DEVIATE GENERATOR
C         IRN = THE RANDOM NUMBER SEED
C         RMU = THE DELAY OF THE NORMAL DISTRIBUTION
C         SIG = THE STANDARD DEVIATION OF THE NORMAL DISTRIBUTION
      ARG=.5*(1.-ABS(1.-2.*URN))
      V=SQRT(-2.*ALOG(ARG))
      PN= C1 + C2*V + C3*V**2
      PD= 1. + C4*V + C5*V**2 + C6*V**3
      A=(URN-.5)/ABS(URN-.5)
      B=V-(PN/PD)
      XNOR= A*SIG*B
      RETURN
C
      FUNCTION RANDOM(I)
```

```
      I=I*131075
      IF(I.LE.0)I=-I
      RANDOM=I*.2910383E-10
      RETURN
      END
```

.SIM4A

BIBLIOGRAPHY

1. Alexsandrov, A. M., "Output Flows from a Class of Queueing Systems", Engineering Cybernetics, No. 1-8, 1968.

2. Box, G. E., and Jenkins, G. M., Time Series Analysis-Forecasting and Control, Holden-Day, San Francisco, 1970.

3. Brown, R. G., and Nilsson, J. W., Introduction to Linear Systems Analysis, John Wiley & Sons, New York, 1962.

4. Cadzow, J. A., Discrete Time Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

5. Daley, D. J., "The Correlation Structure of the Output Process of Some Single Server Queueing Systems", Am. Math. Statist., Vol. 39, Pages 1007-1010, 1968.

6. Daley, D. J., "Notes on Queueing Output Processes", Mathematical Methods in Queueing Theory, Pages 351-358, 1973.

7. Fishman, G. S., Concepts and Methods in Discrete Event Digital Simulation, John Wiley & Sons, New York, 1973.

8. Forrester, J. W., Principles of Systems, Wright-Allen Press, Cambridge, Mass., 1968.

9. Forrester, J. W., Industrial Dynamics, John Wiley & Sons, New York, 1961.

10. Forrester, J. W., Urban Dynamics, MIT Press, Cambridge, Mass., 1969.

11. Forrester, J. W., World Dynamics, Wright-Allen Press, Cambridge, Mass., 1969.

12. Jenkins, G. M., and Watts, D. G., Spectral Analysis, Holden Day, San Francisco, 1968.

13. Kleijnen, J. P. C., Statistical Techniques in Simulation Parts I and II, Marcel Dekher, New York, 1974.

14. Kochenburger, R. J., Computer Simulation of Dynamic Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

15. Krezszig, E., Advanced Engineering Mathematics, John Wiley & Sons, New York, 1967.

16. Mirasol, N. M., "The Output of an M/G/∞ Queueing System is Poisson", Operations Research II, Pages 282-284, 1963.

17. Pritsker, A. B. II, The GASP IV Simulation Language, John Wiley & Sons, New York, 1974.

18. Rektorys, K., Survey of Applicable Mathematics, MIT Press, Cambridge, Mass., 1969.

19. Schweppe, F. C., Uncertain Dynamic Systems, Prentice-Hall, New Jersey, 1973.

20. Schaffer, W. A., "Delays in Dynamic Models", Proceedings, 1974 Summer Computer Simulation Conference, Simulation Councils, LaJolla, California.

21. Shinners, S. M., Modern Control Theory and Applications, Addison-Wesley, Reading, Mass., 1973.

22. Thompson, I. G., "A Simulation Method for Allocating Capital Investment Funds in Airport Baggage/Cargo Handling Systems", Ph.D. Dissertation, University of Texas, 1974.

23. Wright, R. D., "Delay and Smoothing Processes in Dynamic Models", Report Series C-76-1, School of Industrial and Systems Engineering, Georgia Institute of Technology, Jan., 1976.